Server Automation

Software Version: 10.50

Key Concepts Guide

Document Release Date: July, 2016
Software Release Date: July, 2016

Hewlett Packard Enterprise

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notice

Trademark Notices

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: https://softwaresupport.hpe.com/.

This site requires that you register for an HPE Passport and to sign in. To register for an HPE Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HPE Passport login page.

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

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com.

This website provides contact information and details about the products, services, and support that HPE Software offers.

HPE 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 website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HPE 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 HPE Passport user and to sign in. Many also require a support contract. To register for an HPE Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HPE Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the HPESW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/.

Contents

Key concepts

Server Automation (SA) is a data center automation software that centralizes and streamlines many data center functions and automates critical areas of your data center’s server management.

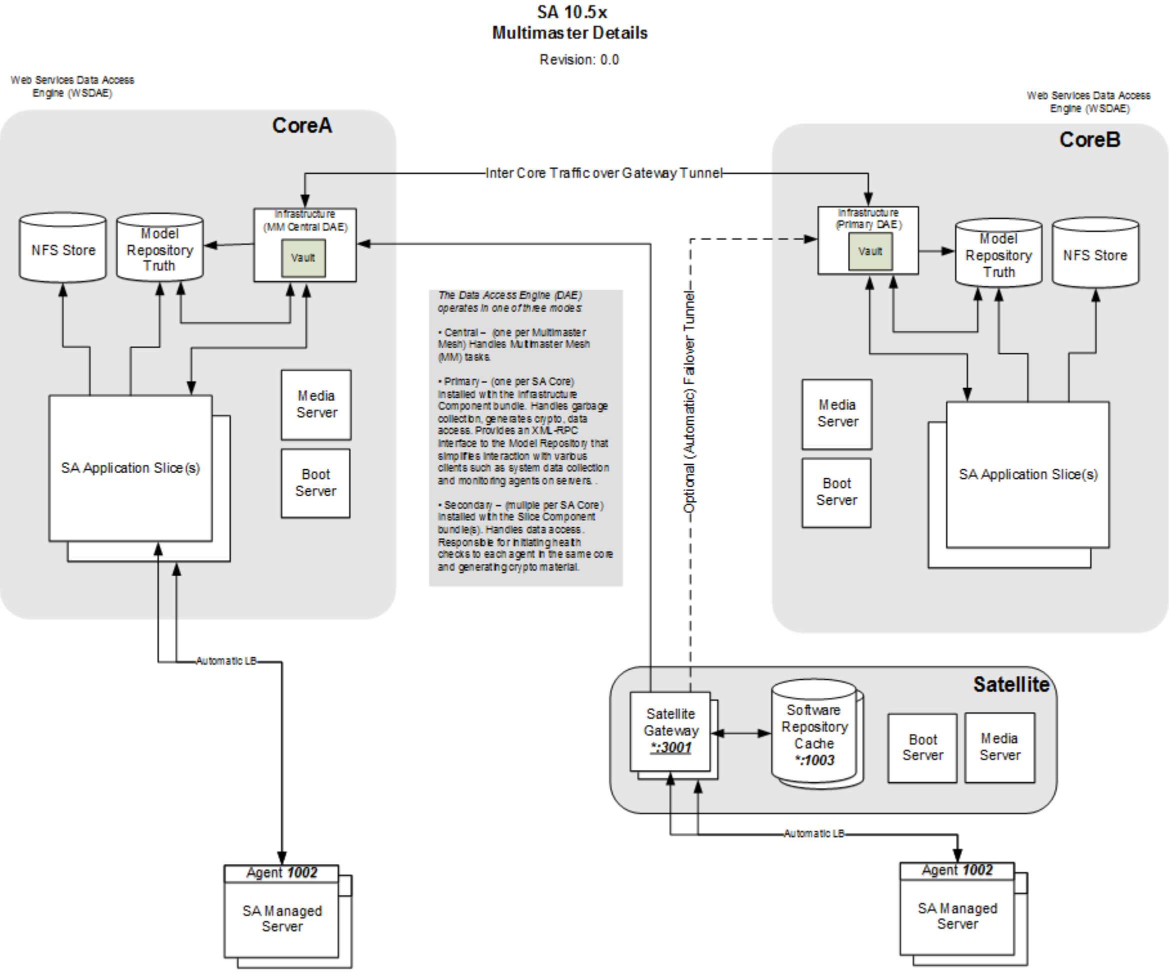
The following topic provides detailed information about SA:

- "Architecture" on the next page
- "Features" on page 34

Architecture

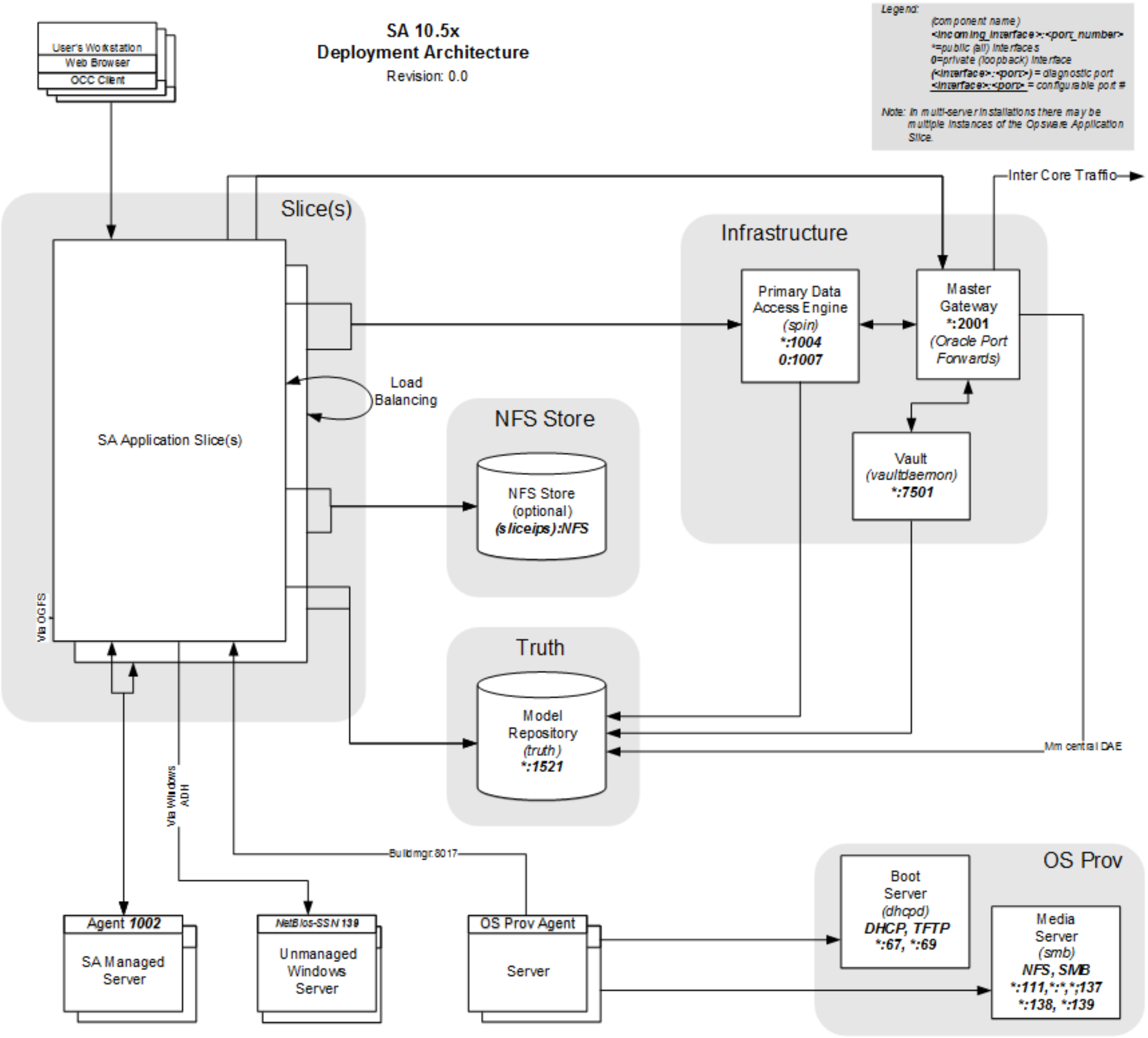
This section is intended for those users who want a more in-depth understanding of SA architecture because they intend to customize the layout of their SA cores, create a Multimaster Mesh, require a remote database installation, and so on. You will learn about the SA Core and its Core Components and the relationship between the core, Server Agents, and Satellites.

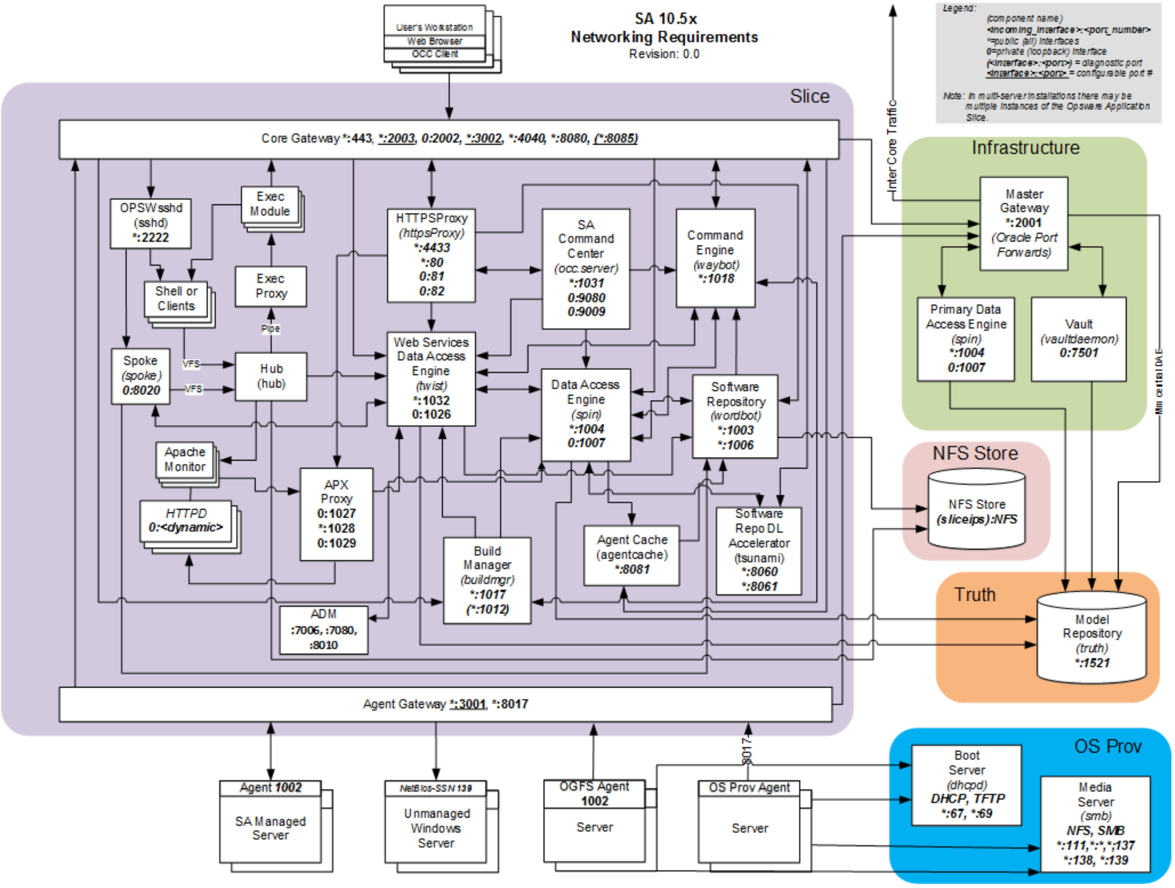
- "SA Core" on page 8
- "SAServer Agents" on page 9
- "Core components" on page 9
- "SA Core component bundling" on page 10
- "Core Component bundles" on page 12



The Data Access Engine (DAE) operates in one of three modes

- Central – (one per Multimaster Mesh) Handles Multimaster Mesh (MM) tasks.
- Primary – (one per SA Core) Installed with the Infrastructure Component bundle. Handles garbage collection, generates crypto, data access. Provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients such as system data collection and monitoring agents on servers. .
- Secondary – (muliple per SA Core) Installed with the Slice Component bundle(s). Handles data access. Responsible for initiating health checks to each agent in the same core and generating crypto material.


User's Workstation
Web Browser
OCC Client
SA 10.5x
Deployment Architecture
Revision: 0.0
Legend:
(component name)
<incoming_interface>:<port_number>
*=public (all) interfaces
0=private (loopback) interface
(<interface>:<port>) = diagnostic port
<interface>:<port> = configurable port #
Note: In multi-server installations there may be multiple instances of the Opsware Application Slice.
Inter Core Traffic
Slice(s)
SA Application Slice(s)
Load Balancing
Via OGFS
Infrastructure
Primary Data Access Engine
(spin)
*:1004
0:1007
Master Gateway
*:2001
(Oracle Port Forwards)
Vault
(vaultdaemon)
*:7501
NFS Store
NFS Store
(optional)
(sliceips):NFS
Truth
Model Repository
(truth)
*:1521
Mm central DAE
Via Windows ADH
Buildmgr:8017
Agent 1002
SA Managed Server
NetBios-SSN 139
Unmanaged Windows Server
OS Prov Agent
Server
OS Prov
Boot Server
(dhcpd)
DHCP, TFTP
*:67, *:69
Media Server
(smb)
NFS, SMB
:111,:*,*;137
*:138, *:139


SA 10.5x
Networking Requirements
Revision: 0.0
User's Workstation
Web Browser
OCC Client
Legend:
(component name)
<incoming_interface>:<port_number>
*=public (all) interfaces
0=private (loopback) interface
(<interface>:<port>) = diagnostic port
<interface>:<port> = configurable port #
Note: In multi-server installations there may be multiple instances of the Opsware Application Slice.
Slice
Core Gateway *:443, *:2003, 0:2002, *:3002, *:4040, *:8080, (*:8085)
Inter Core Traffic
OPSWsshd (sshd) *:2222
Exec Module
Shell or Clients
Exec Proxy
Pipe
HTTPSProxy (httpsProxy) *:4433 *:80 0:81 0:82
SA Command Center (occ.server) *:1031 0:9080 0:9009
Command Engine (waybot) *:1018
Spoke (spoke) 0:8020
VFS
VFS
Hub (hub)
Web Services Data Access Engine (twist) *:1032 0:1026
Data Access Engine (spin) *:1004 0:1007
Software Repository (wordbot) *:1003 *:1006
Apache Monitor
HTTPD 0:<dynamic>
APX Proxy 0:1027 *:1028 0:1029
Build Manager (buildmgr) *:1017 (*:1012)
Agent Cache (agentcache) *:8081
Software Repo DL Accelerator (tsunami) *:8060 *:8061
ADM :7006, :7080, :8010
Agent Gateway *:3001, *:8017
Infrastructure
Master Gateway *:2001 (Oracle Port Forwards)
Primary Data Access Engine (spin) *:1004 0:1007
Vault (vaultdaemon) 0:7501
Mm central DAE
NFS Store
NFS Store (sliceips):NFS
Truth
Model Repository (truth) *:1521
Agent 1002
SA Managed Server
NetBios-SSN 139
Unmanaged Windows Server
OGFS Agent 1002
Server
8017
OS Prov Agent
Server
OS Prov
Boot Server (dhcpd) DHCP, TFTP *:67, *:69
Media Server (smb) NFS, SMB *:111,*:*,*;137 *:138, *:139

SA Core

An *SACore* is a set of *Core Components* that work together to allow you to discover servers on your network, add those servers to a Managed Server Pool, and then provision, configure, patch, monitor, audit, and maintain those servers from an SA Client interface. The SA Client provides a single interface to all the information and management capabilities of SA.

The servers that the Core Components are installed on are called *Core Servers*. Core Components, even if distributed to multiple hosts are still considered part of a single SA Core.

Core Components can all be installed on a single host or distributed across several hosts, however, the typical SA installation uses *Core Component bundling* which installs certain components together on the same server for performance and maintainability purposes. See Core component bundling for more information about component bundling.

To communicate and perform certain server management activities, SA installs *Server Agents* on each Managed Server and communicates with the Managed Servers through Gateways that are part of the

SA Core Components. Server Agents also perform certain actions on Managed Servers as directed by user input from the SA Client.

SAServer Agents

An SA Server Agent is intelligent software that is installed on all servers that you want SA to manage. After an agent is installed on an agentless server, the agent register the server with the SA Core which then adds that server to its pool of Managed Servers. The SA Agent also receives user initiated commands from the Core and takes the appropriate action on the server it is installed on, such as software installation and removal, software and hardware configuration, server status reporting, auditing, and so on.

You can install SA Agents on servers in the following ways:

You can use the SA Agent Deployment Tool (ADT) to discover the servers on your network that do not have SA Server Agents installed (agentless servers) and install agents on those servers. For more information about ADT, see the SA 10.50 User Guide.

You can use SA Provisioning to provision an operating system to a bare-bones server — an SA Server Agent is installed with the operating system. See the SA 10.50 Administration Guide.

You can copy the SA Server Agent binary to the server and install it manually. See the SA 10.50 User Guide.

During agent registration, SA assigns each server a unique ID (the Machine ID (MID)) and stores this ID in the Model Repository. Servers can also be uniquely identified by their MAC Address (the network interface card’s unique hexadecimal hardware identifier, which is used as the device's physical address on the network).

Core components

The Core components are the heart of the SA Core, making it possible to monitor and manage servers. When you retrieve vital information about network servers, provision servers, apply patches, take servers on and off line, configure and audit servers, and more, this interaction is controlled by the Core Components.

The following section describes the SA Core components and interfaces. For detailed information about how the SA Components work together to manage your servers, see the SA 10.50 Administration Guide.

Model Repository

The Model Repository requires either the SA-supplied Oracle database or an existing Oracle installation that meets SA database requirements. For more information about these requirements, see the SA 10.50 Install Guide.

The Model Repository is a standalone component and is not bundled with other Core Components. All SA components work from or update a data model maintained for all SA Managed Servers. The Model Repository stores the following information:

- An inventory of all servers under SA management.
- An inventory of the hardware associated with these servers, including memory, CPUs, storage capacity, and so on.
- Information about managed server configuration.
- An inventory of the operating systems, system software, and applications installed on managed servers.
- An inventory of SA Provisioning operating system installation media (the media itself is stored in the SA Provisioning Media Server).
- An inventory of software available for installation and the software policies that control how the software is configured and installed. The software installation media itself is stored in the Software Repository.
- Authentication and security information.

SA Core component bundling

Certain SA Core Components are *bundled* together and must be installed as a *unit* during a typical installation. It is possible, if necessary, to break certain components (such as the Repository Store, SA Provisioning Media Server, among others) out of a bundle to install them on a different host by performing a Custom installation. However, more complex installations like distributed core components require the services of HPE Professional Services or HPE certified consultants and are not supported for customer installation.

The following table shows the SA component bundles and their constituent components. Note that the Slice Component bundle can have multiple installed instances which aids in workload balancing.

Component distribution

Model Repository	**Infrastructure Components**	**SA Provisioning Components**	**Slice Components #1**	**Slice Components #x**
One per core	One per core	Typically one per core	One per core	Multiple per core
Model Repository	Management Gateway Primary Data Access Engine Model Repository Multimaster Component Software Repository Store (can be located on another host)	Media Server (OS Sequence only) Boot Server (OS Sequence only)	Core Gateway/ Agent Gateway Command Center Global File System Web Services Data Access Engine Secondary Data Access Engine Build Manager Command Engine Software Repository HPE Live Network (HPELN) DCML Exchange Tool (DET) Software Repository Accelerator (`tsunami`) Memcache	Core Gateway/Agent Gateway Command Center Global File System Web Services Data Access Engine Secondary Data Access Engine Build Manager Command Engine Software Repository HPE Live Network (HPELN) DCML Exchange Tool (DET) Software Repository Accelerator (`tsunami`) Memcache

SA Core Component bundling provides the following benefits:

- Added simplicity and robustness for multi-server deployments
- Scaling capability: you can install additional Slice Component bundles for horizontal scaling
- Improved high availability
- Load balancing between slices when multiple instances installed

For more information about SA Core Component architecture and interaction, see the SA 10.50 Key Concepts Guide.

The *Boot Agent* is unrelated to Server Agents and operates as part of SA Provisioning.

Core Component bundles

Infrastructure Component bundle

- **Primary Data Access Engine**
  The Data Access Engine provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients, system data collection, and monitoring agents on servers. The Data Access Engine installed with the Infrastructure Component bundle is designated the *Primary* Data Access Engine. The Data Access Engine installed with the Slice Component bundle(s) is designated the *Secondary* Data Access Engine.

  Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository's schema. The Data Access Engine allows functionality to be added to SA without requiring system-wide changes.

- **Management Gateway**
  Manages communication with other SA Cores and Satellites.

- **Model Repository Multimaster Component**
  The Model Repository Multimaster Component is installed with the Infrastructure Component bundle. A Multimaster Mesh, by definition, has multiple core installations and the Model Repository Multimaster Component synchronizes the data in the Model Repositories for all cores in the Mesh, propagating changes made in one repository to the other repositories.

  Each Model Repository Multimaster Component consists of a Sender and a Receiver. The Sender (Outbound Model Repository Multimaster Component) polls the Model Repository and sends unpublished transactions to other Model Repositories. The Receiver (Inbound Model Repository Multimaster Component) accepts the transactions from other Model Repositories and applies them to the local Model Repository.

- **Software Repository Store**
  The Software Repository Store component can be installed on any server hosting an Infrastructure Component bundle. As of SA 10.50, the Software Repository is part of the Slice Component bundle and the Software Repository Store component has been introduced to handle NFS exports to Slice Component bundle hosts.

  If you choose not to install the Software Repository Store, you must manually configure a NAS (filer) to allow Slice Component bundle servers access to the file system.

  Slice Component bundle

- **Command Engine**
  Part of the Slice Component bundle. The Command Engine is a system for running distributed programs across many servers (typically through SA Server Agents). Command Engine scripts are written in Python and run on the Command Engine server. Command Engine scripts can issue commands to Server Agents. These calls are delivered in a secure manner and are auditable by using data stored in the Model Repository.

  Because you can have multiple Slice Component bundles, and therefore multiple Command Engines, horizontal scaling is greatly enhanced. Multiple Command Engine instances can share the load of command delivery and script execution by taking advantage of the load balancing mechanism provided by multiple Slice Component bundles. Failover and high availability are also improved. For example, when a Command Engine instance tries to delegate a command to another node in the cluster and that node is down, it fails over to the next node.

  SA can use Command Engine scripts to implement functionality.

- **Software Repository**
  Part of the Slice Component bundle. This component is a repository in which the binaries/packages/source for software/application provisioning and remediation is uploaded and stored. A related component is the Software Repository Store which is installed with the Infrastructure Component bundle and handles NFS exports to Slice Component bundle hosts.

  SA supports mirroring of the Software Repository. You can control which Software Repositories in the mesh are designated as mirrors and control the frequency of mirroring jobs by modifying configuration parameters in the SA Client. Mirroring does not affect Satellite Software Repository caches.

  Software Repository mirroring can require large amounts of available disk space. During Standard and Advanced installation, you are given the opportunity to turn off mirroring which is on by default.

  For more information about configuring Software Repository mirroring, see the SA 10.50 Administration Guide.

  For information about how to upload software packages to the SA Library, see the SA 10.50 Administration Guide.

- **Core Gateway/Agent Gateway**
  The Core Gateway communicates directly with Agent Gateways passing requests and responses to and from Core Components.

- **Command Center**

The Command Center (OCC) is the Core Component that underlies the SA Client. The OCC includes an HTTPS proxy server and an application server. You access the OCC only through the SA Client.

- **DCML Exchange Tool**
  The DCML Exchange Tool is installed with each Slice Component bundle and facilitates the import and export of SA content. See the SA 10.50 Administration Guide.

- **Global File System**
  The Global File System (OGFS) is installed with each Slice Component Bundle and provides the central execution environment for SA.

  The OGFS runs on one or more physical servers; customers can scale SA execution capacity by simply adding additional Slice Component bundles in a core.

  The OGFS runs SA built-in components — as well as customer-written programs — within a virtual file system that presents the SA data model, SA actions, and managed servers as virtual files and directories.

  This unique feature of SA allows users of the Global Shell and Automation Platform Extensions (APX) to query SA data and manage servers from any scripting or programming language. Since the OGFS filters all data, actions, and managed server access through the SA security model, programs running in the OGFS are secure by default.

- **Web Services Data Access Engine**
  The Web Services Data Access Engine provides a public-object abstraction layer to the Model Repository and provides increased performance to other Core Components. This object abstraction can be accessed through a Simple Object Access Protocol (SOAP) API, through third-party integration components, or by a binary protocol of components such as the SA Client.

- **Secondary Data Access Engine**
  The Data Access Engine provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients, system data collection, and monitoring agents on servers. The Data Access Engine installed with the Infrastructure Component bundle is designated the *Primary* Data Access Engine. The Data Access Engine installed with the Slice Component bundle(s) is designated the *Secondary* Data Access Engine.

  Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository’s schema. The Data Access Engine allows functionality to be added to SA without requiring system-wide changes.

- **Build Manager**
  (OS Sequences only) Although the Build Manager is part of SA Provisioning, it is installed as part of the Slice Component bundle. The Build Manager facilitates communications between OS

Build Agents and the Command Engine. It accepts SA Provisioning commands from the Command Engine. It provides a runtime environment for the platform-specific build scripts to perform SA Provisioning procedures.

- **HPE Live Network (HPELN)**
  HPE Live Network delivers content updates for Server Automation (SA), Network Automation (NA), Client Automation (CA), Operations Orchestration (OO) and Service Automation Reporter (SAR). The HPE Live Network (HPELN) provides customers with security and compliance policies to help maximize your return on investment in SA, NA, and CA, and to leverage the extensible automation platforms to deliver new automation capabilities on an ongoing basis.

  HPELN is installed as part of the Slice Component bundle during SA Core installation.

- **Software Repository Accelerator** (tsunami)
  An object store download accelerator that boosts remediation performance and scalability for any agents that communicate directly with a Linux-based SA Core.

  Performance and scalability are improved in two key areas:

  RPM Remediation Analysis – Fetching package headers during an RPM dependency analysis/preview is considerably faster than in previous SA releases.

  Remediation Package Staging – Unit downloads to managed hosts from the Software Repository is considerably faster than in previous SA releases and can use 10GbE networking.

- **memcache**
  An in-memory caching layer that works with the Software Repository Accelerator (`tsunami`) component to support remediation and scalability enhancements for agents that communicate directly with a Linux-based SA Core.

  SA Provisioning Components bundle

  - **Boot Server**
    The Boot Server is part of Provisioning. It supports network booting of Sun and x86 systems with inetboot and PXE, respectively. The processes used to provide this support is the Internet Software Consortium DHCP server.

  - **Media Server**
    The Media Server is part of Provisioning. It is responsible for providing network access to the vendor-supplied media used during SA Provisioning. The processes used to provide this support include the Samba SMB server and Linux NFS. You copy and upload your valid operating system installation media to the Media Server.

The OS Build Agent is part of SA Provisioning. It runs during the pre-provisioning (network boot) process and is responsible for registering a server with the SA Core through the Build Manager and guiding the OS installation process.

- **Core Gateway/Agent Gateway**

  The Core Gateway communicates directly with Agent Gateways passing requests and responses to and from Core Components.

- **DCML Exchange Tool**

  The DCML Exchange Tool is installed with each Slice Component bundle and facilitates the import and export of SA content. See the SA 10.50 Administration Guide.

- **Global File System**

  The Global File System (OGFS) is installed with each Slice Component Bundle and provides the central execution environment for SA.

  The OGFS runs on one or more physical servers; customers can scale SA execution capacity by simply adding additional Slice Component bundles in a core.

  The OGFS runs SA built-in components — as well as customer-written programs — within a virtual file system that presents the SA data model, SA actions, and managed servers as virtual files and directories.

  This unique feature of SA allows users of the Global Shell and Automation Platform Extensions (APX) to query SA data and manage servers from any scripting or programming language. Since the OGFS filters all data, actions, and managed server access through the SA security model, programs running in the OGFS are secure by default.

- **Web Services Data Access Engine**

  The Web Services Data Access Engine provides a public-object abstraction layer to the Model Repository and provides increased performance to other Core Components. This object abstraction can be accessed through a Simple Object Access Protocol (SOAP) API, through third-party integration components.

- **Secondary Data Access Engine**

  The Data Access Engine provides an XML-RPC interface to the Model Repository that simplifies interaction with various clients, system data collection, and monitoring agents on servers. The Data Access Engine installed with the Infrastructure Component bundle is designated the *Primary* Data Access Engine. The Data Access Engine installed with the Slice Component bundle(s) is designated the *Secondary* Data Access Engine.

Because interactions with the Model Repository go through the Data Access Engine, clients are less impacted by changes to the Model Repository’s schema. The Data Access Engine allows functionality to be added to SA without requiring system-wide changes.

- **Build Manager**

  (OS Sequences only) Although the Build Manager is part of SA Provisioning, it is installed as part of the Slice Component bundle. The Build Manager facilitates communications between OS Build Agents and the Command Engine. It accepts SA Provisioning commands from the Command Engine. It provides a runtime environment for the platform-specific build scripts to perform SA Provisioning procedures.

- **HPE Live Network (HPELN)**

  HPE Live Network delivers content updates for Server Automation (SA), Network Automation (NA), Client Automation (CA), Operations Orchestration (OO) and Service Automation Reporter (SAR). The HPE Live Network (HPELN) provides customers with security and compliance policies to help maximize your return on investment in SA, NA, and CA, and to leverage the extensible automation platforms to deliver new automation capabilities on an ongoing basis.

  HPELN is installed as part of the Slice Component bundle during SA Core installation.

- **Software Repository Accelerator** (tsunami)

  An object store download accelerator that boosts remediation performance and scalability for any agents that communicate directly with a Linux-based SA Core.

  Performance and scalability are improved in two key areas:

  RPM Remediation Analysis – Fetching package headers during an RPM dependency analysis/preview is considerably faster than in previous SA releases.

  Remediation Package Staging – Unit downloads to managed hosts from the Software Repository is considerably faster than in previous SA releases and can use 10GbE networking.

- memcache

  An in-memory caching layer that works with the Software Repository Accelerator (`tsunami`) component to support remediation and scalability enhancements for agents that communicate directly with a Linux-based SA Core.

SA Provisioning Components bundle

- **Boot Server**

  The Boot Server is part of Provisioning. It supports network booting of Sun and x86 systems with inetboot and PXE, respectively. The processes used to provide this support is the Internet Software Consortium DHCP server.

- **Media Server**

  The Media Server is part of Provisioning. It is responsible for providing network access to the vendor-supplied media used during SA Provisioning. The processes used to provide this support include the Samba SMB server and Linux NFS. You copy and upload your valid operating system installation media to the Media Server.

  The OS Build Agent is part of SA Provisioning. It runs during the pre-provisioning (network boot) process and is responsible for registering a server with the SA Core through the Build Manager and guiding the OS installation process.

Satellite installations

- **Software Repository Cache**
  A Software Repository Cache contains local copies of the contents of a Core’s Software Repository (or of another Satellite). Having a local copy of the Software Repository can improve performance and decrease network traffic when you install or update software on a Satellite’s Managed Servers.

- **Satellite Agent Gateway**
  The Satellite Agent Gateway handles communications between the Satellite and the Core through the Core’s Management Gateway.

SA gateways

SA gateways manage communication between Managed Servers and a SA Core, between multiple cores (Multimaster Mesh), and between Satellite installations and an SA Core. Multimaster installations are discussed in "Multimaster Mesh (Multiple Cores)" on page 21 and Satellite installations are discussed in "SA Satellites" on page 26.

There are several types of gateways:

- **Management Gateway**
  This gateway manages communication between SA Cores and between SA Cores and Satellites.

- **Core Gateway/Agent Gateway**
  These gateways work together to facilitate communication between the SA Core and SA Agents on managed servers.

- **Satellite Gateway**
  This gateway communicates with the Core through the Management Gateway or the Core Gateway depending on your configuration.

Multimaster Master Gateway backup routes

By default, installation of a third or subsequent core in a Multimaster Mesh automatically creates a backup route to the Second Core providing a primary route to the First Core and a backup route to the Second Core. SA creates the Gateway backup routes automatically during installation, you are not required to provide any configuration information, however, if SA cannot create the backup routes, you will see a message to that effect and may need to contact HPE Technical Support to manually configure Gateway backup routes.

Gateway backup routes are created only during fresh installations of SA10.50, not during upgrades. If you are upgrading to SA10.50 from an earlier version, the upgrade will not create gateway backup routes. You must create backup routes manually. Contact your HPE Technical Support representative for more information.

For example, for a three core or more mesh, all Multimaster traffic is routed by default through the First Core’s Master Gateway. However, the Second Core’s Master Gateway is now designated by default as the backup Master Gateway should the First Core’s Master Gateway fail. All additional subsequent core’s Master Gateways added to the mesh will be designated as a backups in the order of installation. On installation, the third and subsequent cores will by default have two tunnels. The first tunnel communicates with the First Core’s Master Gateway, the second tunnel with the second core in the mesh (see the following figure).

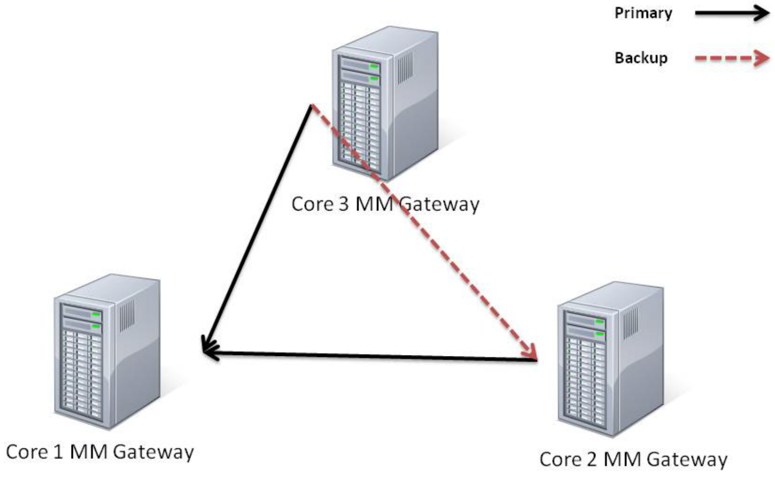


Figure-2: Proposed Setup. 3-Box Core With Backup Route

A mesh with multiple Master Gateways will also have redundant backup routes (see the following figure).

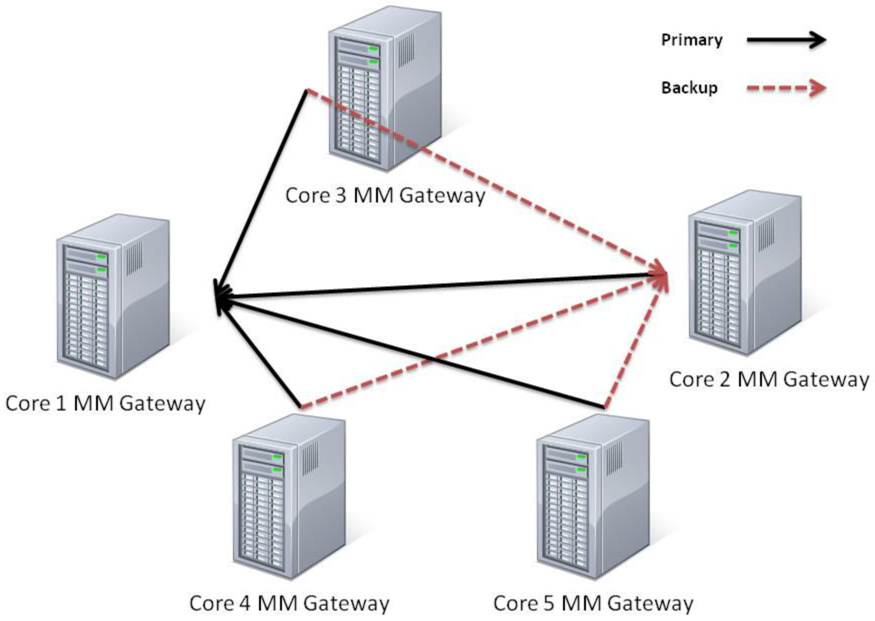


Figure-3: Proposed Setup. 5-Box Core With Backup Route

Upon failure of a Master Gateway, the backup route will automatically be used for Multimaster Mesh traffic by default. When the failed Master Gateway is brought back on line, mesh traffic will automatically be routed through that gateway again.

SA topologies

You must decide what SA topology fits your facility's needs. This section provides some background on the SA topologies to help you make that decision.

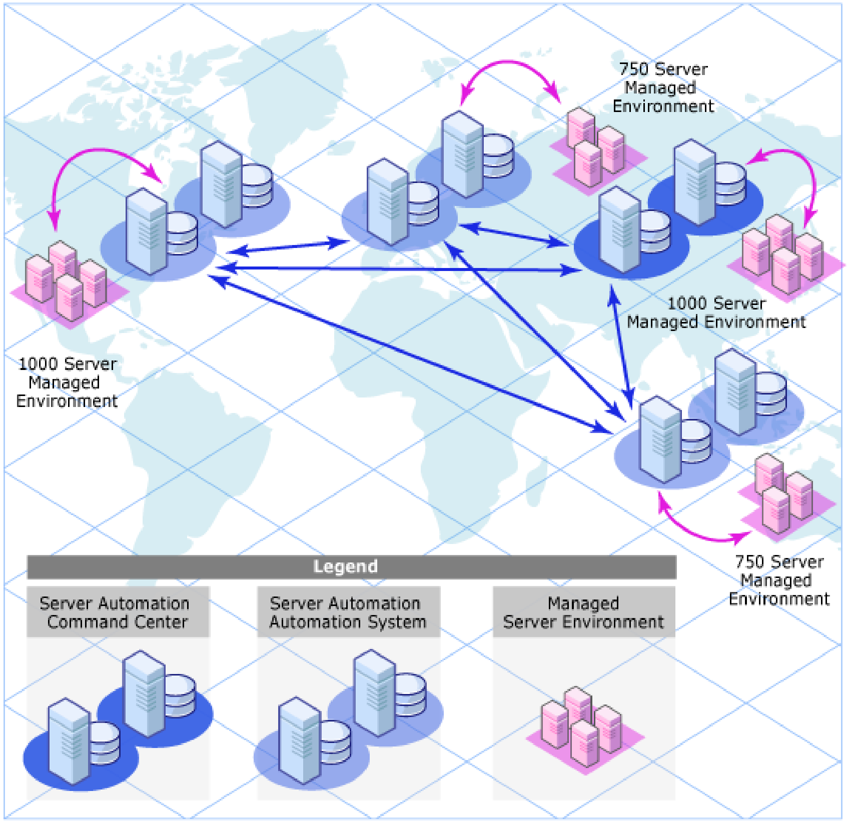
Single Host Core

The simplest topology is a Single Host Core (formerly a Standalone Core) that manages servers in a single facility.

A Single Host Core is best for a small network of servers contained in a single facility. Although a Single Host Core does not communicate with other SA Cores, it has all the components required to do so and can be easily converted into a core that is part of a Multimaster Mesh.

Multimaster Mesh (Multiple Cores)

To manage servers in more than one facility, you install a Multimaster Mesh of SA Cores or a combination of SA Cores and Satellites.


750 Server Managed Environment
1000 Server Managed Environment
1000 Server Managed Environment
750 Server Managed Environment
Legend
Server Automation Command Center
Server Automation Automation System
Managed Server Environment

A *Multimaster Mesh* is a set of two or more SA Cores that communicate with each other through Management Gateways and can perform synchronization of the data about the Managed Servers contained in their respective Model Repositories. Changes to the data in any Model Repository in a Multimaster Mesh are broadcast to all other Model repositories in the Mesh and synchronized.

The SA Core Component that propagates and synchronizes changes from each Model Repository to all other Model Repositories is called the Model Repository Multimaster Component and is part of the Infrastructure Component bundle. This replication capability allows you to store and maintain a “blueprint” of software and environment characteristics for each facility making it easy to rebuild your infrastructure. It also provides the ability to easily provision additional capacity, distribute updates, and share software builds, templates and dependencies across multiple facilities.

A Multimaster Mesh can also include Satellite installations.

Servers can be managed from any facility with an installed SA Core using the SA Client.

Benefits of Multimaster Mesh

An Multimaster Mesh offers the following benefits:

- **Centralized Administration** —The Managed Servers in a Multimaster Mesh can be centrally administered from any facility that is managed by an SA Core that is part of the mesh. Administration is not locked into a single location or even restricted geographically.
- **Redundancy** —Synchronized (replicated) data management between facilities provides redundancy. For example, if an SA Core in one facility in a mesh is damaged, other cores in the Multimaster Mesh contain synchronized copies of the managed server data that can be used to restore the damaged core’s Model Repository to a last known good state. In addition, while a damaged core is unavailable, other cores in the mesh can continue functioning without interruption. Replication also provides the ability to close down or add a facility while other facilities in the mesh continue operations without interruption.
- **Performance Scalability**—In a Multimaster Mesh, only multimaster database synchronizations are transmitted over the network reducing network bandwidth load.
- **Geographic Independence**—Cores can continue to manage servers during network interruptions regardless of location.

Facilities and realms

SA Gateways use two constructs that facilitate routing network traffic and eliminate the possibility of IP address conflicts:

Facilities

A Facility is a construct that represents a collection of servers that a single SA core manages through the data about the managed environment stored in its Model Repository. A facility typically represents a specific geographical location, such as Sunnyvale, San Francisco, or New York, or, commonly, a specific data center.

A Facility is a permissions boundary within SA; that is, a user’s permissions in one Facility do not carry over to another. Every Managed Server is assigned to a single facility. When a device initially registers with the SA Core, it is assigned to the facility associated with the gateway through which it is registering.

For example, Admin A works in Sunnyvale and is in charge of maintaining server patches. In a Facility framework, Admin A is bound to the Sunnyvale Facility as a user. When Admin A views servers, only those servers that are also bound to the Sunnyvale Facility are displayed. He will not see servers for any other Facility.

There are two types of facilities:

- **Core Facilities**: There is one Core Facility for every Core installation.
- **Satellite Facilities**: A default Facility created when you install a Satellite.

Realms

Realms are an SA construct that allow SA to manage servers on different networks in the same Facility without IP address conflicts. A realm is a unique identifier, appended to the IP address of a device in a Facility's network, that allows SA Gateways to uniquely identify devices on different networks in a Multimaster Mesh that may have conflicting IP addresses.

A Realm is a logical entity that defines an IP namespace within which all Managed Server IP addresses must be unique. However, servers that are assigned to different Realms can have duplicate IP addresses and still be uniquely identified within SA by their Realm membership.

Realms are interconnected by gateways in a gateway mesh — a single interconnected network of SA Gateways.

When you create and name a new Facility during installation, a default Realm is also created with the same name as the Facility. For example, when you create the Facility, Datacenter, the installation also creates a Realm named Datacenter. Subsequent Realms in that facility could be named Datacenter001, Datacenter002, and so on. Managed servers in each realm are uniquely identified by the combination of the Realm name and the IP address.

A connection within the mesh has an ingress (or source) realm, where the connection enters the mesh, and an egress (or destination) realm, where the connection exits the mesh.

All SA managed devices are assigned to exactly one realm. Which realm they are assigned to can change dynamically (though it typically will not change). Whenever a device registers with the core, the core looks up the ingress realm for the registration using a gateway identity server, and then assigns the device to that realm.

If the registration occurs using a direct connection, the device is assigned to the transitional realm. The transitional realm may be considered a non-realm. It exists only to denote devices that register themselves with a core using a direct connection.

With the exception of the transitional realm, realms do not span facilities (that is, every realm has a relationship to exactly one facility except for the transitional realm).

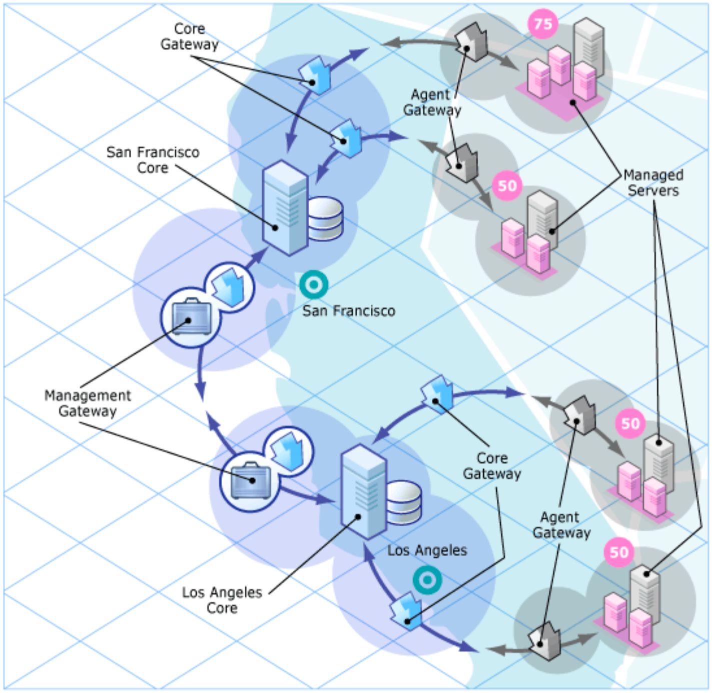
Within the mesh, there are two categories of realms:

- Non-root realms
- Root realms: The mesh has one or more realms at the “center” of the mesh. These realms are root realms. When a gateway is asked to route a connection, if that connection does not specify a destination realm, the connection will be routed to the “closest” root realm, where “closest” is

whichever root realm is reachable at the least network cost. Root realms also have a special relationship in SA to facilities. Whenever an SA Core is installed, a facility is created. At the same time, a root realm is created that has a one-to-one relationship with the new facility.

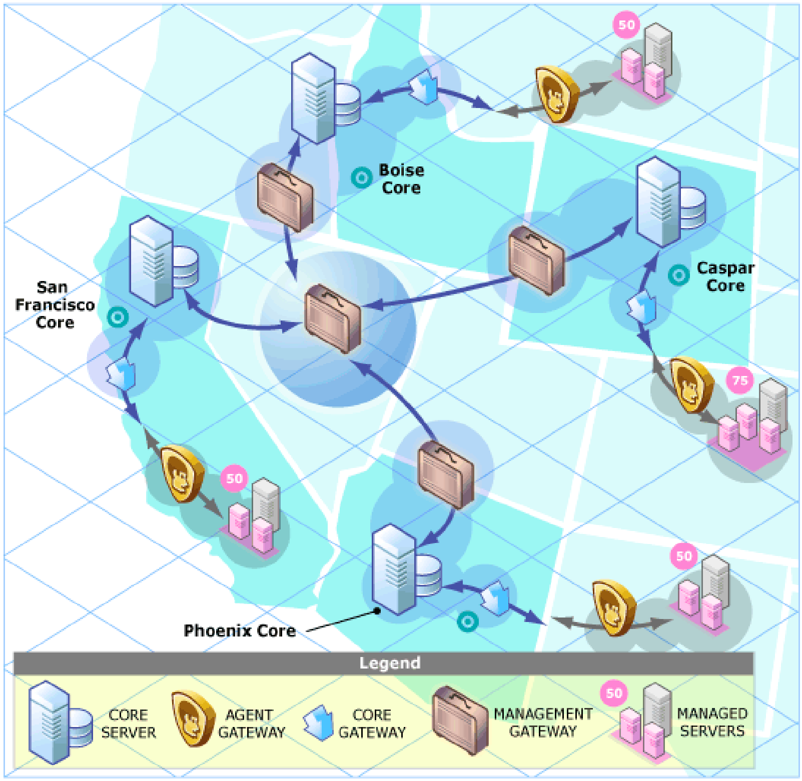
Multimaster Mesh topology examples

The following figure shows a Multimaster Mesh with cores installed in two separate facilities, San Francisco and Los Angeles. Each facility’s core has a Model Repository that contains data about the Managed Servers in both facilities. That data is constantly synchronized (replicated) between both Facilities’ Model Repositories. The cores communicate through their respective Management Gateways.



Communication from the Managed Servers in the Los Angeles facility to the San Francisco core travels through the Los Angeles Agent Gateway to the Core Gateway, then to the Los Angeles Management Gateway, which then communicates with the San Francisco core through the San Francisco Management Gateway and Core Gateway.

The following figure shows a Multimaster Mesh with four cores. This Mesh topology is called a Star Formation with the San Francisco core at the center of the Mesh. The SA Installer configures a Multimaster Mesh in a star topology with backup gateway routes by default.



SA Satellites

A Satellite installation can be a solution for remote sites that do not have a large enough number of potentially Managed Servers to justify a full SA Core installation. A Satellite installation allows you to install only the minimum necessary Core Components on the Satellite host which then accesses the Primary Core's database and other services through an SA Gateway connection.

A Satellite installation can also relieve bandwidth problems for remote sites that may be connected to a primary facility through a limited network connection. You can cap a Satellite's use of network bandwidth to a specified bit rate limit. This allows you to ensure that Satellite network traffic will not interfere with your other critical systems network bandwidth requirements on the same pipe.

A Satellite installation typically consists of, at minimum, a Satellite Gateway and a Software Repository Cache and still allows you to fully manage servers at a remote facility. The Software Repository Cache contains local copies of software packages to be installed on Managed Servers in the Satellite while the Satellite Gateway handles communication with the Primary Core.

You can optionally install the SA Provisioning Boot Server and Media Server on the Satellite host to support remote SA Provisioning. Installing other components on the Satellite host is not supported.

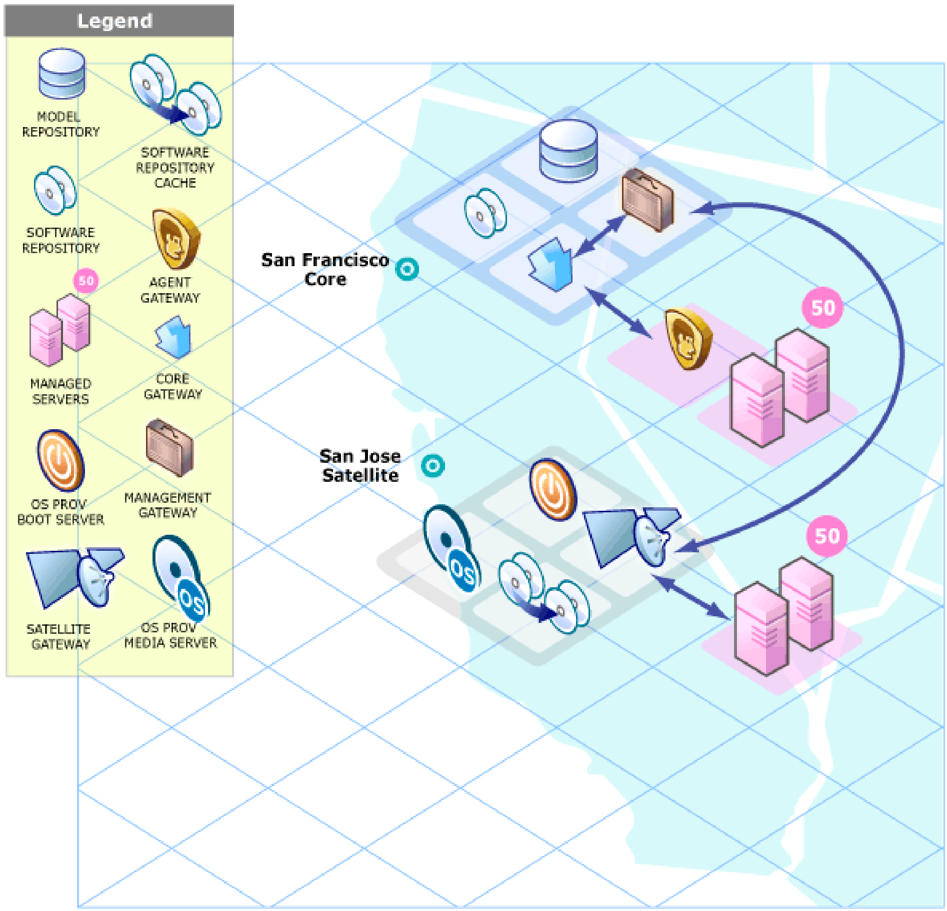
Satellite Topology examples

A simple Single-Core-to-Satellite link

The following figure shows a single Satellite linked to a Single Core. In this example, the main facility is in San Francisco, and a smaller remote facility is in San Jose.

The San Francisco Single Core consists of several components, including the Software Repository, the Model Repository, an Agent gateway, and a Management Gateway. For simplicity, this figure does not show all required Core Components, such as the Command Engine.

The San Jose Satellite consists of a Software Repository Cache, an Satellite Gateway, and an optional SA Provisioning Boot server and Media Server.



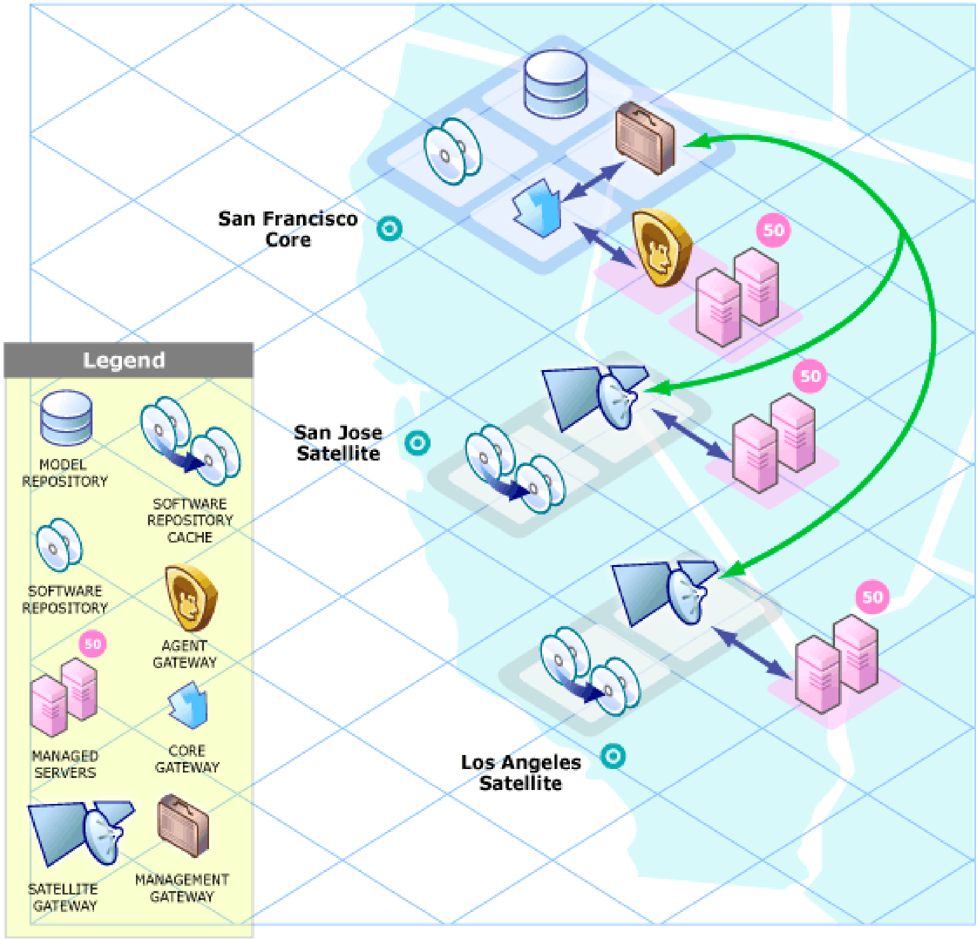
The San Jose Satellite's Software Repository Cache contains local copies of software packages to be installed on Managed Servers in that facility.

The Server Agents installed on managed servers at the San Jose facility connect to the San Francisco core through the San Jose Satellite Gateway, which communicates with the San Francisco Management Gateway, then through the San Francisco Core gateway, ultimately, with the required Core Components.

Return communication reverses that path. The Server Agents installed on managed servers in the San Francisco facility communicate with the Core Components through the San Francisco facility's Agent and Core Gateways.

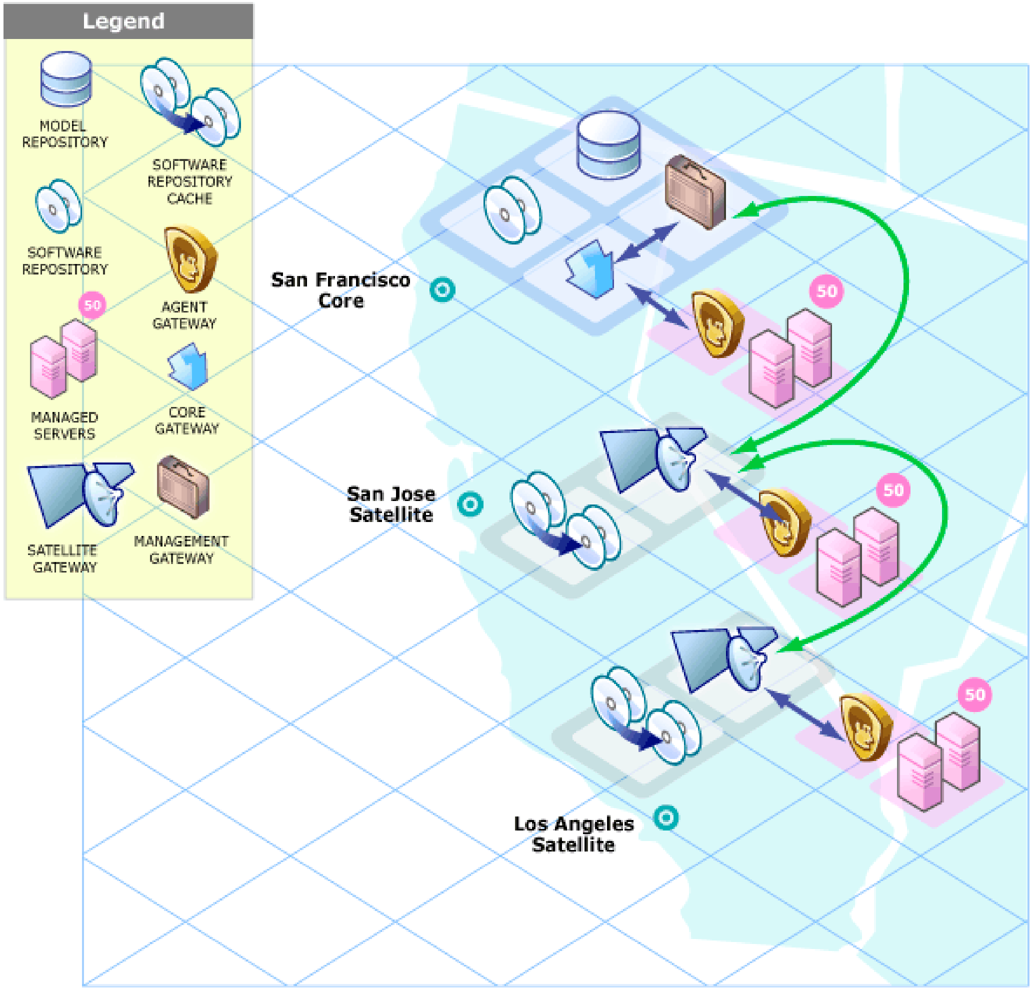
A two-Satellite-to-single-Core link

The following figure shows two Satellites linked to a Single Core. In this example, San Francisco is the main facility, and Sunnyvale and San Jose are Satellite facilities.



A cascading Satellite link

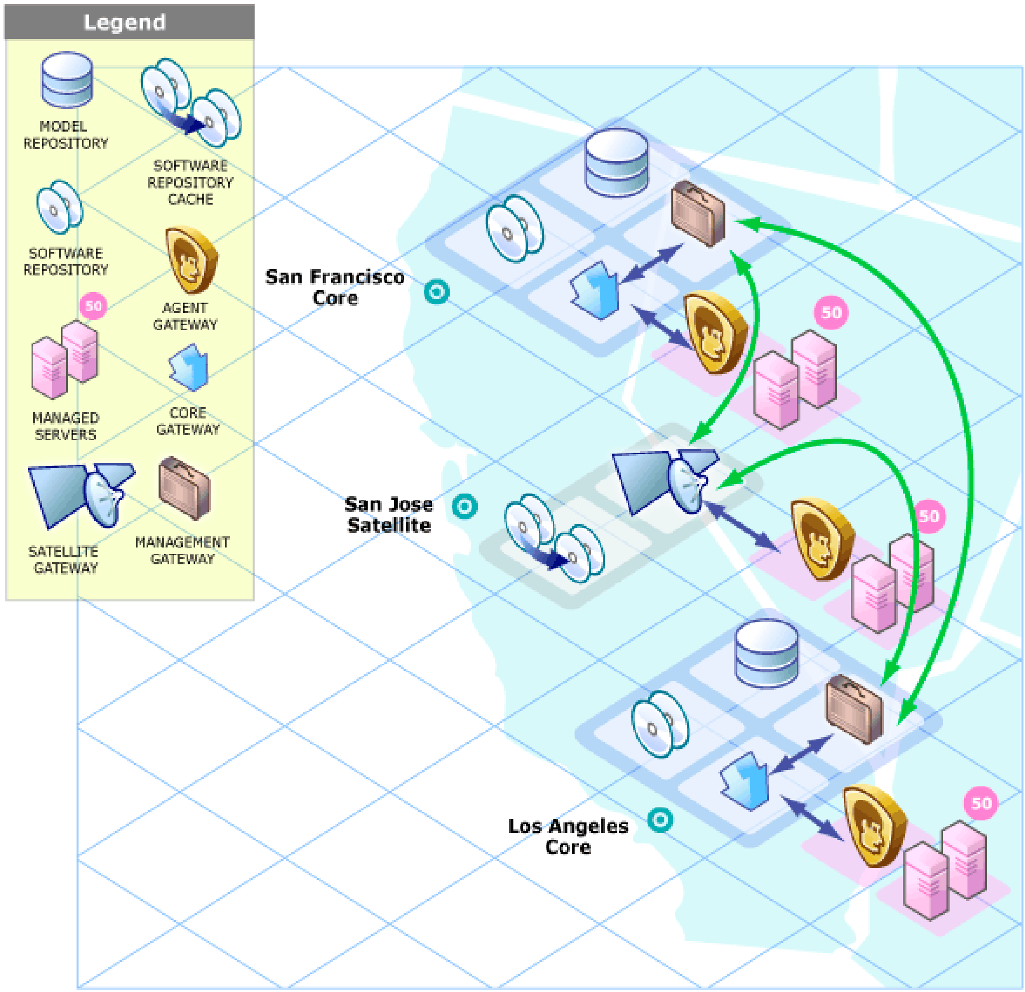
The following figure shows cascading Satellites, a topology in which Satellite Gateways are connected in a chain. This topology enables you to create a hierarchy of Software Repository Caches. Note that the Satellite Gateways in this topology must belong to different SA Realms.



When tasked to install a package on a managed server in the Los Angeles facility, SA first checks to see if the package resides in the Software Repository Cache in Los Angeles. If the package is not in Los Angeles, then SA checks the Software Repository Cache in San Jose. Finally, if the package is not in San Jose, SA goes to the Software Repository in the San Francisco core. For more information, see the SA 10.50 Administration Guide.

Satellites in a Multimaster Mesh

The following figure shows the San Jose Satellite connected to two SA Cores in a Multimaster Mesh.



Even when communication is possible to both Los Angeles and San Francisco, the Management Gateway chooses the route with the lowest cost (in this figure, it is the San Francisco route). You control cost evaluation using a parameter specified during Gateway installation. System designers can specify rules governing which SA Gateway routes to use to minimize network connectivity costs.

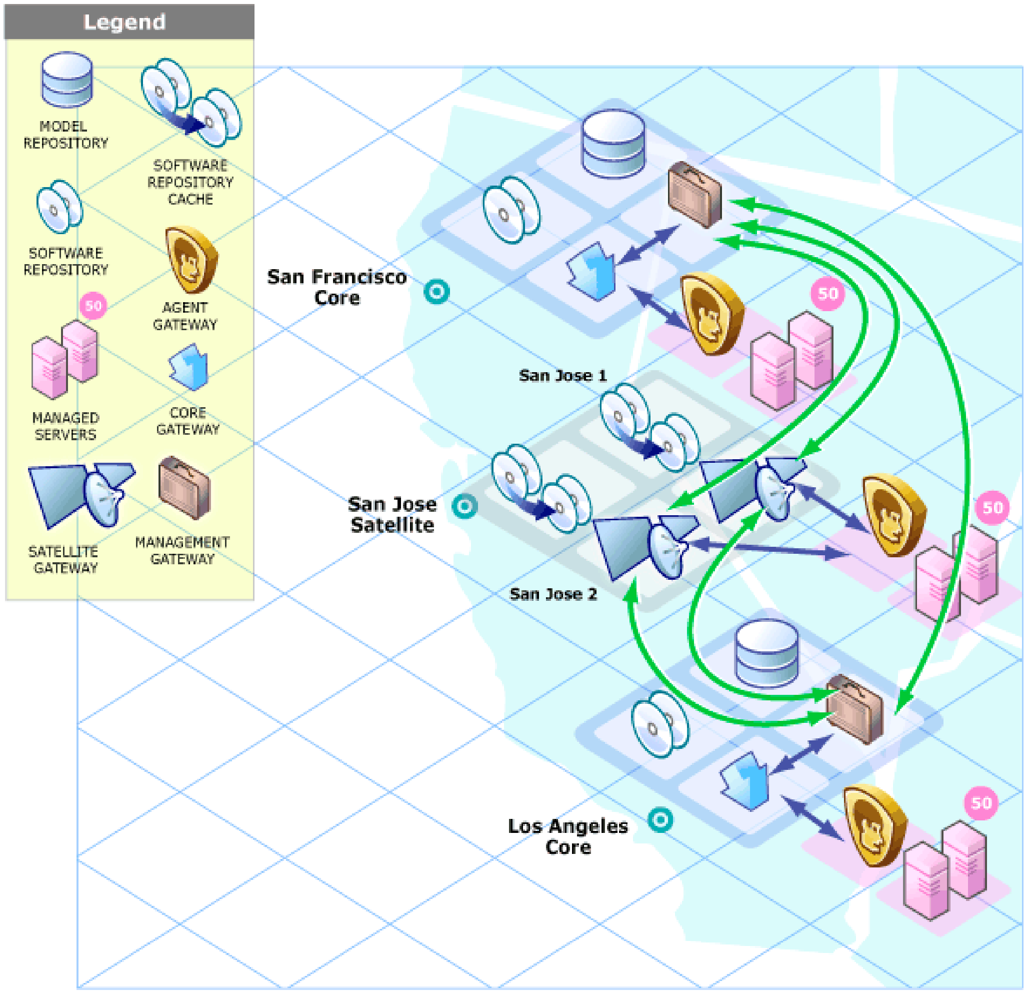
Using the same example environment in a failover scenario, during normal operations, the servers in the San Jose Satellite are managed by the San Francisco Core. Note, however, that the San Francisco and the Los Angeles Cores are directly connected through their Management Gateways.

If the connection between the San Jose Satellite and the San Francisco Core fails, the San Jose Satellite Gateway can move communications immediately from San Francisco to the Los Angeles core, allowing that core to maintain management of the San Jose servers. The Los Angeles Core will have up-to-date information about the San Jose site, because the San Francisco Core's Model

Repository data will have been replicated to the Los Angeles Model Repository as a part of normal SA operations.

Satellite With multiple gateways in a Multimaster Mesh

The following figure shows a topology that provides failover capability in two ways. First, the San Jose Satellites 1 and 2 have Gateway connections to both the San Francisco and Los Angeles Management Gateways. If the Los Angeles core becomes unavailable, the San Francisco core can still manage the servers in the San Jose Satellite.



Second, the Agents installed on the Managed Servers in the San Jose Facility point to both of the Satellite’s Agent Gateways. SA Agents automatically load balance over the available Agent Gateways and therefore can communicate directly with either the San Francisco or Los Angeles cores.

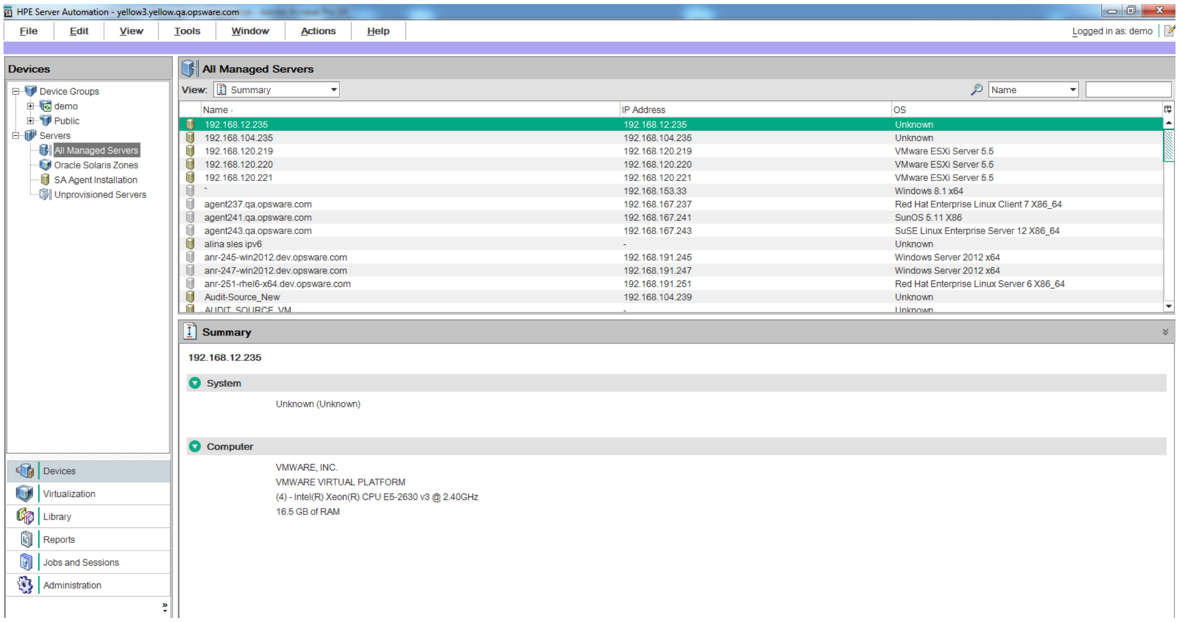
If one Gateway becomes unavailable, the Agents that are using the unavailable gateway as their primary gateway will automatically failover to using the secondary gateway. During routine agent-to-core communication, SA Agents will discover new gateways added to (or removed from) the Satellite.

SA Client

SA Client is a Windows application that you install after SA is installed. It provides an interface to SA functions.

To install the SA Client, you must download and install the SA Client by opening the homepage to your core and clicking **Download Server Automation Client**.

The following figure shows the SA Client main screen. You can find more detailed information about the SA Client in the Using.



Name	IP Address	OS
192.168.12.235	192.168.12.235	Unknown
192.168.104.235	192.168.104.235	Unknown
192.168.120.219	192.168.120.219	VMware ESXi Server 5.5
192.168.120.220	192.168.120.220	VMware ESXi Server 5.5
192.168.120.221	192.168.120.221	VMware ESXi Server 5.5
`	192.168.153.33	Windows 8.1 x64
agent237.qa.opsware.com	192.168.167.237	Red Hat Enterprise Linux Client 7 X86_64
agent241.qa.opsware.com	192.168.167.241	SunOS 5.11 X86
agent243.qa.opsware.com	192.168.167.243	SuSE Linux Enterprise Server 12 X86_64
alina sles ipv6	-	Unknown
anr-245-win2012.dev.opsware.com	192.168.191.245	Windows Server 2012 x64
anr-247-win2012.dev.opsware.com	192.168.191.247	Windows Server 2012 x64
anr-251-rhel6-x64.dev.opsware.com	192.168.191.251	Red Hat Enterprise Linux Server 6 X86_64
Audit-Source_New	192.168.104.239	Unknown
AUDIT_SOURCE_VM	-	Unknown

192.168.12.235

System

Unknown (Unknown)

Computer

VMWARE, INC.
VMWARE VIRTUAL PLATFORM
(4) - Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
16.5 GB of RAM

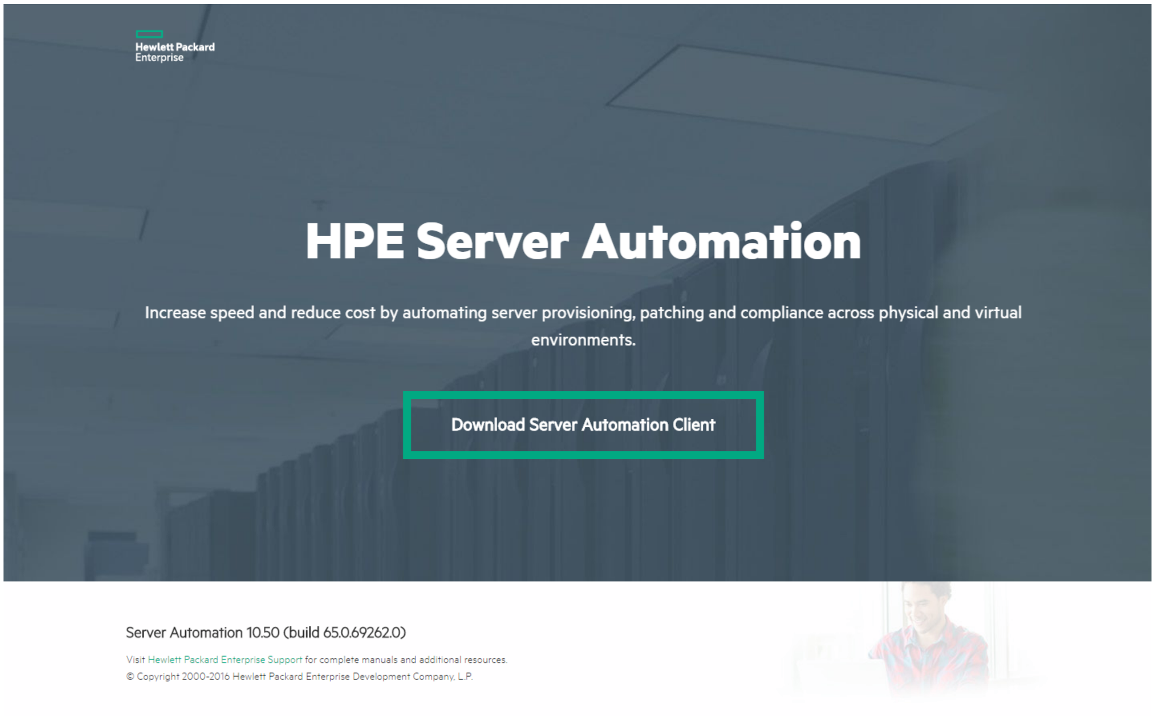
SA Web Client

The SA Web Client is deprecated. Certain SA functions are still provided through the SA Web Client; however, you should use the SA Client when possible.

The SA Web Client is the web-based user interface to SA through which you can download the SA Client launcher. Instructions for starting the SA Web Client in your browser are in the User Guide:

Server Automation. After you start the SA Web Client, you can download and install the SA Client Launcher executable.

The following figure shows the SA Web Client homepage



HPE Server Automation

Increase speed and reduce cost by automating server provisioning, patching and compliance across physical and virtual environments.

Server Automation 10.50 (build 65.0.69262.0)

Visit Hewlett Packard Enterprise Support for complete manuals and additional resources.

© Copyright 2000-2016 Hewlett Packard Enterprise Development Company, L.P.

Features

SA automates data center processes, replacing ad hoc, error-prone, manual processes. For example, by using SAProvisioning, you can set standards for different types of servers and automatically provision the servers, saving time and ensuring that operating system builds are consistent.

You can establish patch polices to install and maintain patches for supported operating systems running on managed servers in your IT environment.

By using Compliance, you have visibility across your managed servers to see which servers are out of compliance. You can then remediate noncompliant servers to bring them back into compliance, based on the policies you created.

SA provides the following capabilities:

- " Device Explorer" on the next page
- "Virtualization management " on the next page
- " Application Configuration Management" on page 36
- "Audit and remediation" on page 36
- "Patch management for Windows" on page 37
- "Patch management for HP-UX" on page 38
- "Patch management for Solaris and Solaris 11" on page 38
- "Patch management for Ubuntu" on page 39
- "Patch management for UNIX" on page 40
- "Reports" on page 41
- "SA Provisioning" on page 41
- "Application Deployment" on page 43
- "Script Execution" on page 44
- " Agentless server discovery and SA agent Installation " on page 45
- "Service Automation Visualizer (SAV)" on page 45
- "Compliance in the SA Client" on page 45
- "Software management" on page 46

- "Global Shell" on page 47
- "FIPS 140-2 compliance" on page 47

SA supports cross-platform environments and is designed to automate both new and existing data center environments.

Device Explorer

The Device Explorer lets you view information about servers in your managed environment.

From the Server Explorer, you can perform the following tasks:

- Create a server snapshot, perform a server audit, audit application configurations, create a package, and open a remote terminal session on a remote server.
- Browse a server’s file system, registry, hardware inventory, software and patch lists, and services.
- Browse SA information such as properties, configurable applications, and even server history.
- From the Groups Browser, you can perform the following tasks:
- Audit system information, take a server snapshot, and configure applications.
- View and access group members (servers and other groups).
- View group summary and history information.

Virtualization management

HPE-supported integrations with virtualization vendors and cloud computing solutions are referred to collectively as virtualization services.

The virtualization vendors manage multiple hypervisors and VMs in a virtualization environment. HPE supports integration with VMware vCenter Server and Microsoft System Center Virtual Machine Manager (SCVMM).

Cloud computing solutions such as OpenStack offer Infrastructure as a Service (IaaS). HPE supports limited integration with OpenStack.

Virtualization management in HPE Server Automationprovides:

- **Visibility** into your datacenter and all your physical and virtual machines (VMs).
- **Compliance** with all your regulatory and enterprise policies.

- **Control** over your entire virtual environment so you can keep VM sprawl in check and detect and resolve problems quickly.

Application Configuration Management

Application Configuration Management (ACM) allows you to create templates so you can modify and manage application configurations associated with server applications. ACM enables you to manage, update, and modify those configurations from a central location, ensuring that applications in your facility are accurately and consistently configured.

Using ACM, you can perform the following tasks:

- Manage configurations based on files and objects, such as Windows registry, IIS metabase, WebSphere, COM+, and more.
- Preview configuration changes before applying them.
- Edit and push configuration changes to individual servers or server groups.
- Use information in the SA data model to set configuration values.
- Manage configurations of any application by building configuration templates.
- Audit the Application Configurations on a server to determine if any of the configuration files on the server are out of sync with the values stored in your templates.

See the SA 10.50 Developer Guide for more information.

Audit and remediation

Audit and Remediation allows you to identify which objects you want checked, where you want to check for them, and when you want to check them in your IT environment.

- *Audit policies* define what to check—such as files, directories, configuration values, and so on.
- *Audits* define where to check—such as servers and server groups.
- *Audit schedules* define when to check—such as one time or as a recurring job.

These capabilities help you understand how to make your managed server environment compliant and how to keep your servers compliant. In SA, you can define server configuration policies to ensure that servers in your facilities meet policy standards. When servers are found to be *out of compliance*—not configured the way you want them to be—you can remediate them to comply with your organization’s standards.

Using the SA Client, you can audit server configuration values based on a live server or a server snapshot, based on your own custom values, or based on pre-configured audit policies. You can also take server configuration snapshots to capture the current state of a system, so that you can compare other servers against a known baseline.

Audit policies allow you to define company or industry-wide compliance and security standards, which can then be used inside of audits, snapshot specifications, and other audit policies. Referencing audit policies in your audits or snapshot specifications helps verify that you are up to date with the latest compliance definitions in your organization.

Using Audit and Remediation, you can perform the following tasks:

- Compare servers or snapshots to reference servers or snapshots
- Create audits for repeated use
- Create audit policies that define compliance and security standards for your organization
- Associate audits with individual servers or dynamic server groups
- Remediate problems at multiple levels, including files, directories, patches, registry keys, and packages

Patch management for Windows

Patch management for Windows enables you to identify, install, and remove Microsoft® Windows patches, and maintain a high level of security across managed servers in your organization. You can identify and install patches that protect against security vulnerabilities for Windows operating systems.

See the SA 10.5 Support and Compatibility Matrix for more information.

Because Windows patches are often released to address security threats, an organization must be able to roll out patches quickly, before systems are compromised. However, at the same time, patches themselves can cause serious problems, from performance degradation to server failures.

While patch management allows you to react quickly to newly discovered threats, it also provides support for strict testing and standardization of patch installation. And, if patches cause problems, even after being tested and approved, Windows patch management allows you to uninstall the patches in a safe and standardized way.

See the SA 10.50 User Guide for more information.

Patch management for HP-UX

SA automates HP-UX Patch management by enabling you to:

- Define HP-UX software policies that provide a model-based approach to managing your HP-UX servers. Server Automation enables you to create a model of your IT environment using HP-UX software policies. These software policies specify patches and scripts that can be installed on the managed servers.
- Install HP-UX patches and patch bundles on your managed servers.
- Establish a patch installation process.
- Schedule the stages of patch management: analysis, download, and installation. You can also set up email notification for each stage and associate a ticket ID for each job.
- Verify the compliance status of servers, based on software policies.
- Display the Compliance view to see whether servers are configured according to the software policy and to remediate non-compliant servers.
- Search for software resources and servers.
- Use the SA Library to search for HP-UX packages, patches, and software policies using powerful and flexible search criteria, such as availability, architecture, operating system, reboot options, version, and so on. You can also search for HP-UX software policies by name, folder name, availability, and operating system.
- View patch dependencies and patch applicability analysis while previewing patch installation.

See the SA 10.50 User Guide for more information.

Patch management for Solaris and Solaris 11

Server Automation patch management for Solaris enables you to identify, install, and remove Solaris patches, and maintain a high level of security across managed servers in your organization.

Server Automation patch management for Solaris allows you to automate the process of installing and uninstalling Solaris patches and patch clusters on Sun Solaris using patch policies. In addition, SA analyzes the dependency, supersedence, and applicability relationships between patches in the policy and displays an updated and ordered list of patches that should be installed on the server. This feature allows you to verify the compliance status of a server and remediate non-compliant servers and automatically download the Solaris patches into SA and organize them into patch policies.

SA automates Solaris patching by enabling you to:

- Determine which patches your managed servers need.
- Create Solaris patch policies.
- Download Solaris patches, patch clusters, and patch bundles, and then store them, and related vendor information, in the SA Library.
- Resolve all dependent patches for Solaris patches.
- Install Solaris patches and patch clusters on managed servers.
- Install Solaris patches in single-user mode.
- Install patches by Oracle Solaris zones.
- Establish a patch installation process.
- Verify the compliance status of servers with patch policies.
- Search for software resources and servers.

See the SA 10.50 User Guide for more information.

Patch management for Ubuntu

HPE Server Automation patch management for Ubuntu enables you to identify, install, and remove Ubuntu Debian package updates, and maintain a high level of security across managed servers in your organization. You can identify and install Ubuntu packages that protect against security vulnerabilities for the SA-supported Managed Server platforms.

SA automates the key aspects of patch management while offering a fine degree of control over how and under what conditions Ubuntu packages are installed. By automating the patching process, patch management can reduce the amount of downtime required for patching. SA also allows you to schedule patch activity, so that patching occurs during off-peak hours.

With Ubuntu patching in SA, you can import the metadata before importing the binary packages. You can run the Ubuntu scanner with only the metadata downloaded to determine the server vulnerabilities. Then you can run the Ubuntu package importer to import only those packages that are required by managed servers. This practice saves you storage space as well as scan and remediation processing time.

The Ubuntu Patch Management documentation contains information about how to import Ubuntu metadata and packages, scan for vulnerabilities, and install Ubuntu package updates using patch policies.

SA automates Ubuntu patching by providing the following features and capabilities:

- **A central repository** where packages are stored and organized in their native formats.
- **A database** that stores information about every package that has been applied.
- **Dynamic Patch Policies t**hat analyze platform vulnerabilities based on the latest metadata from the vendor.
- **Advanced search** abilities that identify servers that require package updates.
- **Auditing abilities** for tracking the deployment of important package updates.

See the SA 10.50 User Guide for more information.

Patch management for UNIX

Patch Management for UNIX enables you to identify, install, and remove patches, to maintain a high level of security across managed servers in your organization. Using the SA Client, you can identify and install patches that protect against security vulnerabilities for AIX operating systems.

SA allows you to react quickly to newly discovered security threats and also provides support for strict testing and standardization of patch installation. If patches cause problems after being tested and approved, SA allows you to uninstall the patches in a safe and standardized way.

SA stores patch information in the SA Library that includes detailed information about every server under management, the patches and software installed on the servers, and the patches and software available for installation. You can use this data to determine the severity of your exposure to a newly discovered threats, and to help assess the benefits of rolling out a patch versus the costs in downtime and testing requirements.

By automating the patching procedure, SA can reduce the amount of downtime required for patching. SA also allows you to schedule patch activity, so that patching occurs during off-peak hours.

Patch Management for UNIX provides the following capabilities that enable you to browse patches by a certain operating system, schedule patch downloads and installations, set up email notifications, preview a patch installation, use software policies and remediation to install and uninstall patches, and export patch information to a reusable file format:

- The SA Library where patches are stored and organized in their formats
- A database that includes information on every patch that has been applied
- Customized scripts that can be run before and after a patch is installed
- Advanced search abilities that identify servers that require patching
- Auditing abilities that enable security personnel to track the deployment of important patches

See the SA 10.50 User Guide for more information.

Reports

SA Reports provide comprehensive, real-time information about managed servers, network devices, software, patches, customers, facilities, operating systems, compliance policies, and users and security in your environment. These reports are presented in graphical and tabular format, and are actionable—where you can perform appropriate actions on objects, such as a policy or an audit, within the report. These reports are also exportable to your local file system (in `.html` and `.xls` formats) to facilitate use within your organization.

SA Provisioning

SA Provisioning provides the ability to install (or provision) pre-configured operating system baselines onto bare metal and virtual servers quickly, consistently, and with minimal manual intervention. Bare metal and virtual server SA Provisioning is a key part of the overall process of getting a server into production.

SA Provisioning ensures that each server in your facility has a standardized, default operating system configuration that you control. For detailed information about SA Provisioning, see the SA 10.50 Administration Guide.

Benefits of SA Provisioning include:

- **Integration with other SA functionality**
  Because SA Provisioning is integrated with the suite of SA automation capabilities, including patch management, software management, and distributed script execution, hand-offs between IT groups are seamless. SA ensures that all IT groups are working with a shared understanding of the current state of the environment, which is an essential element of delivering high-quality operations and reliable change management.
- **Update server baselines without re-imaging** Unlike many other provisioning solutions, systems provisioned with can be easily changed after provisioning to adapt to new requirements. The key to this benefit is the use of templates and an installation-based approach to provisioning.
- **Flexible architecture designed to work in many environments** Provisioning supports many different types of servers, networks, security architectures, and operational processes. works well in CD (Linux provisioning) or network-boot environments (both DHCP and non-DHCP environments), with scheduled or on-demand workflows, and across a large variety of hardware

models. This flexibility ensures that you can provision operating systems to suit your organization’s needs.

You can perform SA Provisioning functions from the SA Client. SA automates the entire process of provisioning a comprehensive server baseline, which typically consists of the following tasks:

- Preparing the hardware for OS installation using an OS installation profile (required only for OS Sequences).
- Defining OS Build Plans which are a list of tasks to be performed on a server before and after operating system installation. OS Build Plans are more powerful than and can be used in place of OS Sequences.
- Defining OS Sequences which are a list of tasks to be performed on a server during installation. OS Sequences can include application, patch, and remediation policies. SA recommends that you use the more flexible OS Build Plans.
- Installing a base operating system and default OS configuration using an OS Build Plan or OS Sequence.
- Applying the latest set of OS patches. The exact list depends on the applications running on the server.
- Executing pre-installation or post-installation scripts that configure the system with values such as a root password.
- Installing system agents and utilities such as SSH, PC Anywhere, backup agents, monitoring agents, or anti-virus software.
- Installing widely shared system software such as Java Virtual Machines.

SA Provisioning supports:

- Windows, Solaris, and Linux.
- Network or CD/DVD-based installations.
- Separation of duties between data center staff and systems administrators.
- A model-based approach — in which you create a *standard build* in SA which can then be installed on many systems.

SA Provisioning integrates with your operating system vendors’ native installation technology, specifically:

- Windows setup answer files: `unattend.xml, sysprep.inf`
- Red Hat Kickstart

- SuSE YaST (Yet another Setup Tool)
- Solaris Jumpstart
- WINPE/WIN-BCOM/UNDI

You can provision an operating system on:

- A server in SA’s agentless server pool that does not have an operating system installed (bare metal server)
- A virtual server
- A server in SA’s unmanaged server pool with an installed operating system
- A server in SA’s managed server pool with an installed operating system (reprovisioning)

Application Deployment

With Application Deployment, you can create, test, and deploy your custom software applications to target servers in your data centers. For example, you can move applications from the development team to the quality assurance team for testing. Once testing is complete, you can move the application to other phases such as preproduction, staging, and finally to production. The Application Deployment tool reduces the complex communications necessary to deploy applications by providing a single point of access where everyone involved can view or enter data that is relevant to them and to their role.

With Application Deployment, you can:

- Model your application components such as code, scripts, configuration files, and tiers such as application servers, web servers, and databases.
- Manage multiple concurrent releases and versions of your applications.
- Deploy, roll back, and undeploy your applications on target servers.
- Model your target servers that are running the tiers required by your applications. These target servers are managed servers in Server Automation.
- Provide clear, concise communication between software application developers, Quality Assurance, and testing, systems administrators, and other operations personnel.
- Model and implement life cycles from application development to QA to preproduction to staging to production. You can customize SA to match your enterprise life cycle.

For complete information, see the SA 10.50 Developer Guide.

Script Execution

SA Script Execution enables you to share and run ad-hoc or saved scripts across an entire farm of SA-managed servers.

By executing scripts with SA instead of manually, administrators benefit from:

- Parallel script execution across many UNIX and/or Windows servers, saving time and ensuring consistency.
- Role-based access control, ensuring only authorized administrators can execute scripts on hosts to which they have access.
- The ability to control access to scripts by storing them in private or in public libraries.
- The ability to see and download script output one server at a time or in a consolidated report, which captures output from all servers in a single place.
- The ability for scripts to be mass-customized. Administrators can access information in SA about the environment and the state of servers. This is critical to ensuring that the right scripts are executed on the right servers.
- A comprehensive audit trail that reports who, what, when, and where a particular script was executed.

Because Script Execution is an integrated part of SA, administrators can take advantage of unique benefits when compared to standalone script execution tools:

- Using known system state and configuration information to customize script execution, users can tailor each script by referencing and accessing the rich store of information in SA, such as the customer or business that owns the server, whether the server is a staging or production server, which facility the server is located in, and custom name-value pairs.
- By sharing scripts without compromised security, users can share scripts with each other without compromising security because SA maintains strict controls on who can execute scripts on which servers and generates a comprehensive audit trail of script execution.

Agentless server discovery and SA agent Installation

Agentless server discovery and SA agent installation allows you to deploy Server Agents to a large number of servers in your facility and place them under SA management.

You can perform the following tasks:

- Scan your network for agentless servers.
- Select servers for SA Agent installation.
- Select a communication tool and provide user/password combinations.
- Choose agent installation options and deploy agents.

Service Automation Visualizer (SAV)

Service Automation Visualizer (SAV) is designed to help you optimally understand and manage the operational architecture and behavior of distributed business applications in your IT environment. Since these applications are complex collections of services that typically run across many servers, as well as network, it can become increasingly difficult to understand (or remember) what is connected to what, where performance problems originate, how to troubleshoot and resolve problems, and what result would occur if you make a change in your environment.

SAV helps you see (visualize) this type of information through physical and logical drawings.

Compliance in the SA Client

In the SA Client, the Compliance view allows you to see the overall compliance levels for all servers and groups of servers in your facility. From this view, which is commonly known as the *compliance dashboard*, you can remediate servers that are *out of compliance*. You can view compliance for an individual server, multiple servers, groups of servers, or for all servers under SA management.

The compliance dashboard displays the results of all compliance statuses on servers or groups of servers for audits, audit policies, software policies, patch policies, and application configurations. A server’s compliance status is based on a *compliance policy*. A compliance policy defines unique server configuration settings or values to ensure that your IT environment is configured as it should be.

A compliance policy is typically created and defined by a *policy setter*. In some environments, a system administrator might be required to create an ad-hoc policy. The policy setter creates compliance policies and then attaches them to servers to ensure that servers are compliant with your organization’s standards and policies.

For example, a policy setter can create a software policy that defines a standard set of patches and packages that must be installed on a server. The policy setter can also define the manner in which certain application files must be configured on a server. A server or group of servers is considered compliant if its configuration matches the rules, defined by the policy setter, in the compliance policy.

The compliance dashboard allows you to determine whether the server’s actual installed software, packages, patches, and configuration files settings match the configuration defined in the *software policy*. The Compliance view allows you to view compliance for groups of servers, showing a compliance status rollup for all members and sub-group members of a group. From the Compliance view, you can discover servers and groups of servers that are *out of compliance* and then remediate any problems.

Software management

SA Software Management provides a powerful mechanism to model software by using software policies and to automate the process of deploying software and configuring applications on a server in a single step. In addition, SA Software Management provides a structure to organize your software resources in folders and define security permissions around them. SA Software Management allows you to verify the compliance status of a server and remediate non-compliant servers.

SA Software Management provides the following capabilities:

- Creating an organizational structure for software
- Defining security boundaries for folders
- Defining a model-based approach to manage the IT environment in your organization
- Enabling sharing of software resources among user groups
- Deploying and configuring applications simultaneously
- Deploying multiple application instances on one server
- Establishing a software deployment process
- Verifying compliance status of servers to software policies
- Generating reports
- Comprehensively searching for software resources and servers

See the SA 10.50 User Guide for further information.

Global Shell

The SA Global Shellenables you to manage servers by using a command-line interface. You can remotely perform the following tasks:

- Complete routine maintenance tasks on managed servers.
- Troubleshoot, identify, and remediate problems on managed servers.

The Global Shell consists of a file system and a command-line interface to that file system for managing servers in SA. The file system is known as the SA Global File System (OGFS). All object types in the OGFS (such as servers, customers, and facilities) are represented as directory structures in this file system.

The SA Global Shell also manages user permissions for accessing the file system, Windows Registry, and Windows Services objects on managed servers.

FIPS 140-2 compliance

HPE Server Automation (SA) complies with the Federal Information Processing Standards publication 140-2, a security standard that enables government entities to procure equipment that uses validated cryptographic modules.

This section describes how SA Core, Satellite and managed servers complies with FIPS 140-2 and the methods used to make SA FIPS 140-2 compliant.

- "SA Core" on the next page
- "SA Agent" on the next page
- "SA Gateway" on page 49
- "SA Satellite" on page 49
- "SA Managed server" on page 50

SA Core

An SA Core is set of Core Components that work together to allow you to discover servers on your network, add those servers to a Managed Server Pool, and then provision, configure, patch, monitor, audit, and maintain those servers from a centralized SA Client interface. The SA Client provides a single interface to all the information and management capabilities of SA.

The servers that the Core Components are installed on are called Core Servers. Core Components, even if distributed to multiple hosts are still considered part of a single SA Core. Core Components can all be installed on a single host or distributed across several hosts, however, the typical SA installation uses Core Component bundling which installs certain components together on the same server for performance and maintainability purposes.

In order to communicate and perform certain server management activities, SA installs Server Agents on each Managed Server and communicates with the Managed Servers through Gateways that are part of the SA Core Components. Server Agents also perform certain actions on Managed Servers as directed by user input from the SA Client.

> **Note:** In FIPS mode, sufficient entropy stemming from the character device /dev/random must be available on the core servers, to ensure proper startup and functionality of SA components.

SA Agent

An SA Agent is intelligent software that is installed on a server that you want to manage using SA. After an agent is installed on an unmanaged server, it registers with the SA Core which can then add that server to its pool of Managed Servers. The SA Agent also receives commands from the Core and initiates the appropriate action on its local server, such as software installation and removal, software and hardware configuration, server status reporting, auditing, and so on.

During agent registration, SA assigns each server a unique ID (the Machine ID (MID)) and stores this ID in the Model Repository. Servers can also be uniquely identified by their MAC Address (the network interface card's unique hexadecimal hardware identifier, which is used as the device's physical address on the network).

SA Gateway

SA Gateways manage communication between Managed Servers and an SA Core, between multiple cores, and between Satellite installations and an SA Core.

There are several types of gateways:

- Management Gateway
  This gateway manages communication between SA Cores and between SA Cores and Satellites.
- Core Gateway/Agent Gateway
  These gateways work together to facilitate communication between the SA Core and Agents.
- Satellite Gateway
  This gateway communicates with the SA Core through the Management Gateway or the Core Gateway depending on your configuration.

SA Satellite

A Satellite installation can be a solution for remote sites that do not have a large enough number of potentially Managed Servers to justify a full SA Core installation. A Satellite installation allows you to install only the minimum necessary Core Components on the Satellite host which then accesses the Primary Core’s database and other services through an SA Gateway connection.

A Satellite installation can also relieve bandwidth problems for remote sites that may be connected to a primary facility through a limited network connection. You can cap a Satellite’s use of network bandwidth to a specified bit rate limit. This allows you to insure that Satellite network traffic will not interfere with your other critical systems network bandwidth requirements on the same pipe.

A Satellite installation typically consists of, at minimum, an Satellite Gateway and a Software Repository Cache and still allows you to fully manage servers at a remote facility. The Software Repository Cache contains local copies of software packages to be installed on Managed Servers in the Satellite while the Satellite Gateway handles communication with the Primary Core.

SA Managed server

An SA Managed Server is a server that has an installed SA Agent and is actively under SA management.

Related topics

- "About FIPS 140-2"
- "FIPS 140-2-Compliant Technologies"
- "Supported SA Core and Satellite operating systems "
- "Supported managed server operating systems"
- "Supported FIPS 140-2 security level"
- "Acronyms"
- "Related industry documentation"

About FIPS 140-2

The Federal Information Processing Standards Publication (FIPS) 140-2, “Security Requirements for Cryptographic Modules,” was issued by the National Institute of Standards and Technology (NIST) in May, 2001. The standard specifies the security requirements for cryptographic modules utilized within a security system that protects sensitive, but unclassified information. FIPS 140-2 is one of the standards adopted by the governments of the U.S. and Canada to promote the use of validated cryptographic modules and provide Federal agencies with a security metric to use in procuring equipment that complies with the standard and contains validated cryptographic modules.

HPE Server Automation (SA) supports FIPS 140-2 by using FIPS-compliant cryptographic modules.

FIPS 140-2-Compliant Technologies

SA achieves FIPS 140-2 compliance by using cryptographic modules that have already gone through the NIST certification process. SA uses the following FIPS 140-2-compliant technologies.

NSS cryptographic module

SA employs the FIPS 140-2 certified Network Security Services (NSS) cryptographic module, an open-source, general purpose cryptographic library under the Mozilla Public License.

The NSS cryptographic module contains an API based on the industry standard Public-Key Cryptography Standards (PKCS) #11 cryptographic token interface version 2.20 published by RSA, the security division of EMC Corporation.

TLS/SSL transport protocol

SA also makes use of Transport Layer Security (TLS), the next generation of Secure Sockets Layer (SSL).

The SA platform is composed of multiple distributed components that communicate sensitive information over insecure networks. SSL is a proven industry standard that provides:

- Encryption to ensure that data (events/user interaction) cannot be sniffed
- Data integrity (MAC) to prevent intentional or accidental data modification on the wire
- Authentication to prevent credentials from leaking across the wire

Because the function of TLS and SSL is the same, the protocols are referred to jointly as TLS/ SSL, although they use different algorithms to establish secure key exchange.

The SSL 2.0 and 3.0 protocols are not FIPS 140-2 compliant. TLS is the only SSL variant that incorporates FIPS 140-2-approved algorithms based upon Internet Engineering Task Force (IETF) standards.

SHA-1/SHA-2 family

The Secure Hash Algorithm is a set of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS). SA uses SHA-256, but SHA-1 and other hash functions from SHA-2 family are supported.

Supported SA Core and Satellite operating systems

FIPS 140-2 enabled SA Cores are supported on all supported SA managed platforms.

Supported managed server operating systems

FIPS 140-2 enabled managed servers are supported on all supported SA managed platforms except for the following:

- Red Hat Enterprise Linux 5 on IA 64
- Red Hat Enterprise Linux 5 and 6 on S390X platform (Z Series)
- SUSE Linux Enterprise Server 10 and 11 on S390X platform (Z Series)
- HPUX PA-RISC 11.11, 11.23, 11.31
- HPUS IA64 11.11, 11.23, 11.31
- Windows Server 2008 R2 on IA 64

Supported FIPS 140-2 security level

FIPS 140-2 Security Level

SA Component	Supported FIPS 140-2 Security Level	NSS Version	OpenSSL Version
SA 10.10 and later	Level 1	3.15.1	1.0.1h (2.0.5 FIPS module)

SA cryptography modes

SA offers two cryptographic modes:

- FIPS 140-2 mode (sensitive, but unclassified information)
- ESM Standard Cryptography (default mode)

FIPS 140-2 mode

FIPS 140-2 mode enables security for information that is sensitive, but unclassified (SBU). FIPS 140-2 mode means that the NSS cryptographic module has been deployed and enabled on all the relevant SA components that connect to and exchange data with the SA Core.

FIPS 140-2 mode is based on RSA public-key encryption technology, and is a separate and secure cryptography system apart from ESM's standard cryptography system. Once FIPS 140-2 mode is enabled, ESM's standard cryptography system is not used.

ESM standard cryptography

To support deployments for which FIPS 140-2 cryptography is not a requirement, SA continues using its existing cryptographic algorithms and key store formats.

Acronyms

Acronyms	Abbrevation
ESM	Enterprise Security Management
FIPS	Federal Information Processing Standards
HMAC	Keyed-Hash Message Authentication Codes
HTTPS	Secure Hypertext Transfer Protocol (over TLS/SSL)
ECDSA	Elliptical Curve Digital Signature Algorithm. Used for support of Suite B security for information classified up to top secret.

IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
MD5	Message-Digest Algorithm 5
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSS	Network Security Services
PKCS	Public Key Cryptography Standards
RSA	A public-key encryption technology developed by RSA Security, Inc., the Security Division of EMC Corporation. The acronym stands for Rivest, Shamir, and Adelman, the inventors of the technique.
SBU	Sensitive But Unclassified. Refers to information to be protected by a cryptographic method.
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer; related to TLS
TSL	Transport Security Layer; the next generation of SSL
W3C	World Wide Web Consortium

Related industry documentation

Refer to the following industry resources for more about the FIPS 140-2 standard, and the OpenSSL cryptographic module and its underlying technology.

Related industry documentation

Topic	Resource
FIPS PUB 140-2	Information Processing Standards (FIPS) document published by the Information Technology Laboratory of the National Institute of Standards and Technology (NIST). Issued May 25, 2001. http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
OpenSSL Cryptographic Module	The FIPS 140-2 Non-Proprietary Security Policy Level 1 and 2 Validation for the OpenSSL Cryptographic Module version 0.9.8j, published by OpenSSL.org.

Related industry documentation, continued

Topic	Resource
	http://www.openssl.org/docs/fips/fipsvalidation.html http://www.openssl.org/docs/fips/UserGuide-2.0.pdf
Approved Cryptographic Modules	A list of all cryptographic modules approved by NIST. http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140valall.htm
PKCS #11	A description of the Public Key Cryptographic Standards (PKCS) #11. Describes the cryptographic token interface API, which allows device independence and resource sharing among multiple applications that access multiple devices. http://www.rsa.com/rsalabs/node.asp?id=2133
Transport Layer Protocol (TLS)	An overview of Transport Layer Protocol (TLS), the next generation of Secure Sockets Layer, (SSL). http://en.wikipedia.org/wiki/Transport_Layer_Security Notes about how and why TLS is implemented: Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations published by NIST in 2005: http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf
Internet Engineering Task Force (IETF)	An overview of the IETF, the organization that developed of the TLS protocol, which promotes Internet standards recognized by the World Wide Web Consortium (W3C), International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC): http://en.wikipedia.org/wiki/IETF The IETF web site: http://www.ietf.org/

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Key Concepts Guide (Server Automation 10.50)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to hpe_sa_docs@hpe.com.

We appreciate your feedback!