# 5620 SAM
# SERVICE AWARE MANAGER
# 14.0 R8

# System Architecture Guide

**3HE-10706-AAAF-TQZZA**

**Issue 1**

**March 2017**

# Contents

# About this document

## Purpose

The *5620 SAM System Architecture Guide* is intended for technology officers, network planners, and system administrators to increase their knowledge of the 5620 SAM software structure and components. It describes the system structure, software components, and interfaces. In addition, 5620 SAM fault tolerance, security, and network management are described from an architectural perspective.

## Document support

Customer documentation and product support URLs:

- Customer Documentation Welcome Page
- Technical support

## How to comment

Documentation feedback

# 1 5620 SAM system architecture

## 1.1 5620 SAM system architecture overview

### 1.1.1 Introduction

The 5620 SAM is a network management system that simplifies routine operations and allows the bulk provisioning of network objects. The system is designed using industry standards such as Java, XML/SOAP, WebDAV, and 3GPP. The 5620 SAM uses open-standard interfaces that allow the system to interoperate with a variety of other network monitoring and management systems.

### 1.1.2 5620 SAM functions

The 5620 SAM network management functions include the following:

- service and routing configuration using distributed policies and profiles
- equipment, service, and customer inventory reporting
- network performance, accounting, and flow-based statistics collection
- hierarchical alarm correlation between objects
- interoperation with other network systems

### 1.1.3 Main architecture features

The main features of the 5620 SAM system architecture include the following:

- the use of open standards to promote interaction with other systems
- distributed resources that spread the processing load across multiple components and efficiently execute network management tasks
- a multi-layer design model with functions in separate modules that interact with OEM products to accommodate increasing network growth and complexity
- web services that provide access to 5620 SAM applications by effectively exporting XML interfaces over the Internet; the web services permit access to remote components such as web portals, and allow third-party vendors to create customized entry points for 5620 SAM functions
- component redundancy that provides a high degree of fault tolerance

## 1.2 Network management functions

### 1.2.1 Introduction

The 5620 SAM provides comprehensive network access for operators based on role-based scopes of command and spans of control over types of network objects.

A 5620 SAM system collects data from managed NEs and collates the data for accounting, performance monitoring, troubleshooting, inventory, and fault management. The system deploys operator commands to the network, and performs functions such as NE discovery and configuration backups.

A 5620 SAM system is primarily designed to manage proprietary devices. However, you can obtain drivers for managing some devices from other vendors. Drivers can be downloaded from the customer support site, and driver installation and usage documentation is available from the Customer Documentation Welcome Center.

### 1.2.2 Service management

The 5620 SAM service management functions allow network operators to provision VLL, VLAN, VPLS, IES, VPRN and mirror services for customers. Each service can be monitored to provide performance, usage, and fault information.

### 1.2.3 Accounting

The 5620 SAM collects accounting statistics from managed NEs that can be used to bill subscribers. The statistics are transferred to the 5620 SAM main and auxiliary servers using FTP or SCP.

### 1.2.4 Equipment management

The 5620 SAM maintains an equipment data model and deploys configuration updates to the managed NEs. For example, when a 5620 SAM operator adds a card to an NE, the data model is updated to include the card, and the card provisioning and configuration commands are sent to the NE. New NEs can be discovered at operator request, or automatically. A newly discovered NE is added to the data model.

### 1.2.5 Performance management

The 5620 SAM can monitor services and network resources using performance statistics, OAM diagnostic tools, and data validation, and raises alarms when appropriate.

- The 5620 SAM collects NE performance statistics using SNMP.
- The 5620 SAM has a comprehensive suite of OAM tools for monitoring service, NE, and transport availability and performance. You can run tests before service activation to ensure that a service functions correctly after activation.
- The 5620 SAM regularly compares the configuration information on managed NEs with the information in the 5620 SAM database to ensure synchronization.

### 1.2.6 Fault management

The 5620 SAM performs fault management in response to NE SNMP traps. The system converts traps to status updates and raises alarms when appropriate. GUI clients use visual and auditory cues to alert an operator when an alarm is raised.

The 5620 SAM immediately forwards fault information as JMS events to OSS clients that subscribe to the appropriate JMS topic, and in response to OSS client XML requests for information.

## 1.3 System components

### 1.3.1 Introduction

A 5620 SAM system comprises several components that are described below. Some components are supported only in specific deployment types. See the *5620 SAM Planning Guide* for comprehensive deployment information.

### 1.3.2 Main server

A main server is the central Java-based network-management processing engine that can be collocated on the same station as a 5620 SAM database, or on a separate station. A main server includes third-party components such as an application server, JMS server, web server, protocol stack, and database adapter. Some functions, for example, statistics collection, can be distributed across optional auxiliary servers.

### 1.3.3 Auxiliary server

An auxiliary server, like a main server, is a Java-based processing engine, but is an optional, scalable component that extends the system ability to perform functions such as statistics, PCMD, or call-trace data collection. An auxiliary server is controlled by a main server, and collects data directly from NEs.

### 1.3.4 Cflowd auxiliary server

A Cflowd auxiliary server is an optional, scalable component that collects AA Cflowd statistics directly from NEs and forwards the data to an OSS or third-party application for billing, traffic analysis, or data analytics.

### 1.3.5 Analytics server

An analytics server uses business intelligence software to analyze raw and aggregated statistics data and create reports about various network conditions and trends. The on-demand and scheduled reports are available from the 5620 SAM Analytics application.

An analytics server is deployable only in a 5620 SAM system that includes an auxiliary database, which acts as the analytics data store.

### 1.3.6 5620 SAM database

The 5620 SAM database is a customized relational database that provides persistent storage and serves as a central network data repository. The database can be collocated on the same station as a main server, or on a separate station.

### 1.3.7 Auxiliary database

An auxiliary database is an optional, horizontally scalable database that expands the 5620 SAM storage capacity for demanding operations such as statistics collection. An auxiliary database is a distributed database that is deployed in a cluster of at least three separate stations. Load balancing and data replication among the stations in the cluster provide high performance and robust fault tolerance.

An auxiliary database is deployable only in a 5620 SAM system that includes one or more auxiliary servers.

### 1.3.8 Single-user GUI client

A single-user GUI client is a Java-based graphical interface for network operators. Single-user GUI client deployment is supported on multiple platforms.

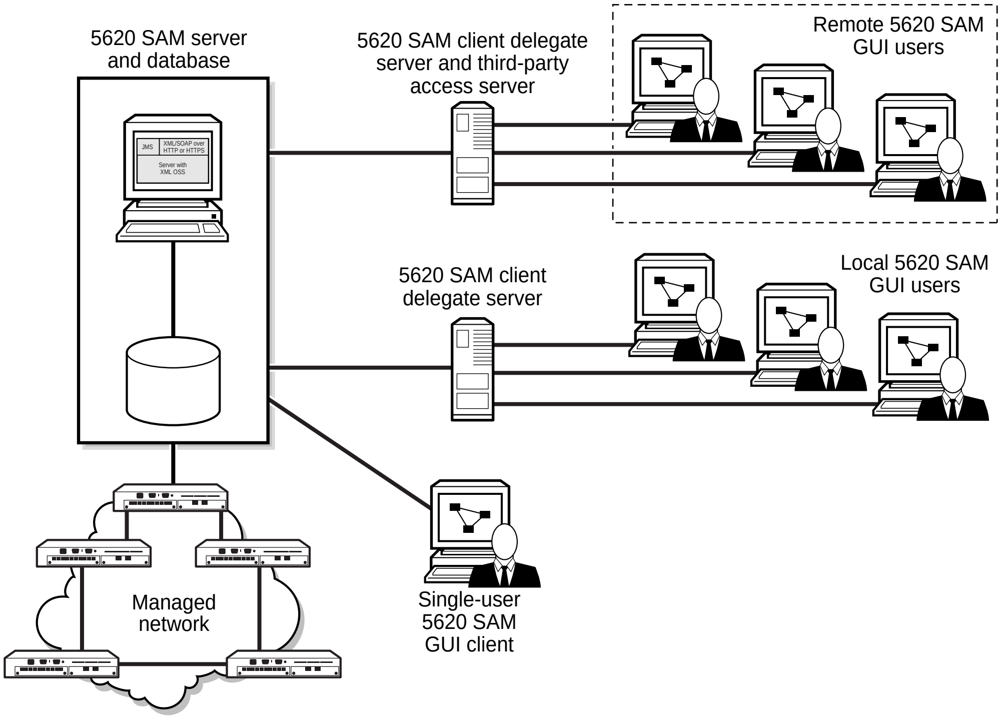
### 1.3.9 Client delegate server

A client delegate server supports simultaneous GUI sessions using one client software installation. A client delegate server can host local and remote user sessions, and supports the use of a third-party remote access tool such as a Citrix gateway. Client delegate server deployment is supported on multiple platforms.

A client GUI session that is opened through a client delegate server is functionally the same as a single-user client session. The client delegate server locally stores the files that are unique to each user session, such as the client logs and GUI preference files, using a directory structure that includes the RHEL or Windows username.

Figure 1, “Client delegate servers” (p. 9) shows two client delegate servers in a 5620 SAM management network. Multiple local users log in to a client delegate server directly, and remote users log in through a client delegate server that hosts a third-party access tool, for example, a Citrix gateway. Another local user opens a session on a single-user client station.

*Figure 1* Client delegate servers



If a client delegate server becomes unreachable, the 5620 SAM raises an alarm and changes the color of the associated session entries in the GUI. The alarm clears when the server is again reachable.

You can use the client software on a client delegate server from the local console. It is recommended that you install a client delegate server, rather than a single-user client, to facilitate the deployment of additional clients.

A main server monitors the registered client delegate servers and displays information about them in the GUI. To register a client delegate server, you specify the client delegate server IP address and installation location during main server installation, upgrade, or configuration.

You can use a client GUI to list the following:

- registered client delegate servers and the availability of each
- active client delegate server sessions
- active client sessions on a specific client delegate server
- active client sessions for a specific 5620 SAM user

The number of allowed 5620 SAM client sessions on a client delegate server is configurable as a threshold using the 5620 SAM GUI. If a user tries to open a client session that reaches or exceeds the threshold, the session proceeds and the client delegate server raises an alarm. This threshold-crossing function can help to balance the session load across multiple client delegate servers. You require the Update user permission on the Server package to configure the threshold.

The following restrictions apply to client delegate servers.

- The installation of only one client delegate server on a station is supported.
- You cannot change a 5620 SAM single-user client to a client delegate server.
- A client delegate server connects to one release of main server; multiple main servers to which the client delegate server connects must be at the same release.
- Depending on the platform type, specific installation, upgrade, and operation security restrictions apply; see the *5620 SAM* **|** *5650 CPAM Installation and Upgrade Guide* for the deployment requirements and restrictions.

### 1.3.10 OSS clients

An OSS client is a software application that you create and implement to automate GUI client tasks, or to retrieve data for post-processing, for example, rolling up statistics data into a billing application. OSS clients may be as diverse as simple CLI scripts and third-party applications. An OSS client is platform-independent, because only Java messages are exchanged with the 5620 SAM.

The 5620 SAM supports the following OSS client types:

- XML/SOAP clients that use the XML OSS interface to perform general network management; XML schema files provide the data object definitions and describe the object attributes and methods; see the *5620 SAM XML OSS Interface Developer Guide* for more information
- 3GPP CORBA or XML/SOAP clients that use the 3GPP OSS interface to perform LTE management; IRPs provide the 3GPP data object definitions; see the *5620 SAM Wireless OSS Interface Developer Guide* for more information

### 1.3.11 Subcomponents

All subcomponents, for example, Java modules, database software, and web server software, are represented by license files in the following directory on a main server:

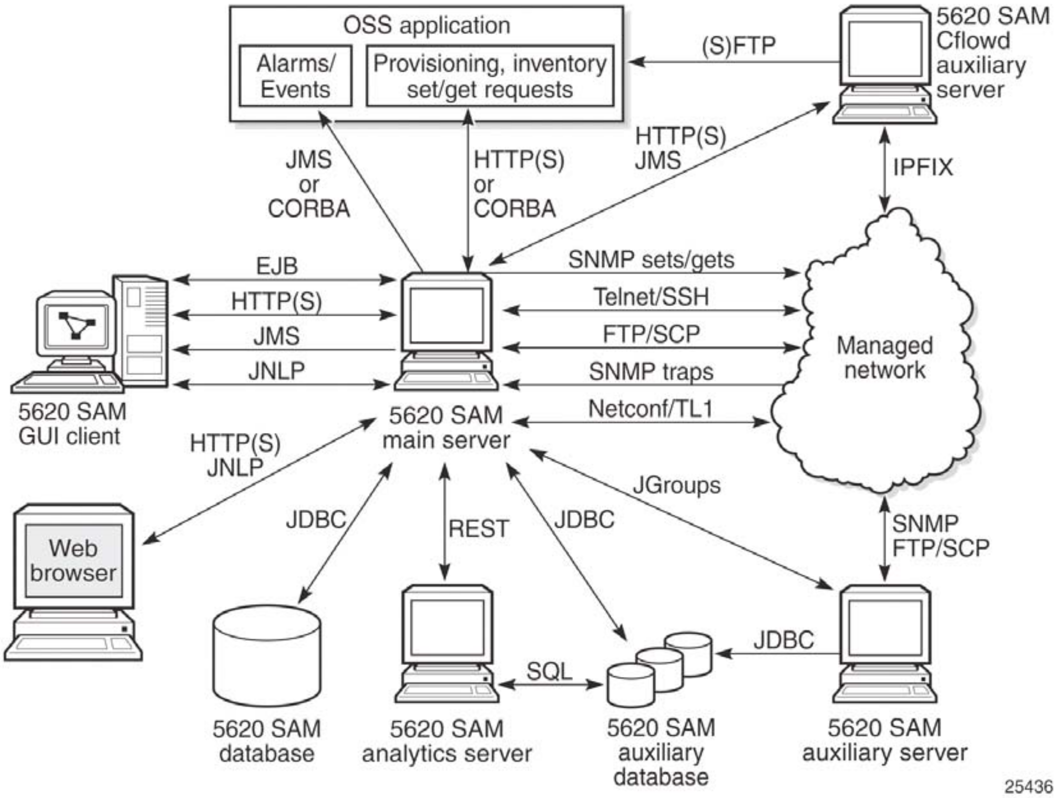
/opt/5620sam/server/nms/distribution/licenses

## 1.4 Component communication

### 1.4.1 Introduction

The 5620 SAM component interfaces use industry-standard protocols for communication between servers, databases, NEs, and clients, as shown in Figure 2, “5620 SAM component communication” (p. 10). All 5620 SAM components communicate with all other 5620 SAM components using IPv4 or IPv6 exclusively; however, a 5620 SAM system can communicate with and manage a network using both protocols concurrently.

*Figure 2* 5620 SAM component communication



### 1.4.2 Servers and managed NEs

Main and auxiliary servers send messages to the managed network in the form of SNMP, FTP, secure FTP, and SCP commands. A 5620 SAM main server also sends CLI commands using Telnet or SSH.

- A main server uses SNMP to monitor and manage network performance, and to identify network problems. Main servers deploy configuration changes to NEs using SNMP. Auxiliary servers poll MIB performance statistics on the NEs, or collect PCMD or call-trace data. The NEs use asynchronous SNMP messages called traps to notify the 5620 SAM of events. UDP streaming is used by NEs for operations such as forwarding PCMD records to the 5620 SAM.
- The CLI of a managed NE is accessible from the client GUI using Telnet or SSH.
- FTP and SCP are transport-layer protocols for transferring files between systems. The 5620 SAM uses the protocols to back up NE configuration data, collect NE accounting statistics, and download software to NEs.

### 1.4.3 Main server and clients

Client interfaces provide access to a 5620 SAM system and the managed network through a main server.

A main server and clients communicate in the following ways:

- GUI clients send requests to the server EJB session beans using Java RMI.
- The GUI client update function uses HTTP or HTTPS for client software updates and file downloads.
- 5620 SAM application clients use HTTP or HTTPS to communicate with the web service on a main server.
- A web-based GUI client communicates through a browser using JNLP.
- XML OSS clients send requests for processing by a main server, and subscribe to JMS topics to receive real-time event notifications. The messages between a main server and an XML OSS client are in XML/SOAP format, and are sent over HTTP or HTTPS. The JMS and the XML publisher service on a main server run in separate JVMs to support multiple concurrent client connections. See the *5620 SAM XML OSS Interface Developer Guide* for more information about the messaging between XML OSS clients and main servers.
- 3GPP OSS clients send CORBA or SOAP/XML requests over HTTP or HTTPS for processing by a main server. See the *5620 SAM Wireless OSS Interface Developer Guide* for information about the messaging between 3GPP OSS clients and main servers.

### 1.4.4 Main server and 5620 SAM database

A main server communicates with a 5620 SAM database using a JDBC session over TCP. JDBC is a Java API for interworking with SQL relational databases.

### 1.4.5 Main server and auxiliary servers

A main server includes a mechanism for sending requests to auxiliary servers. An auxiliary server notifies the main server after it finishes processing a request. If the main server fails to send a request, or all auxiliary servers are unresponsive to a request, the main server raises an alarm.

### 1.4.6 Main server and Cflowd auxiliary servers

A Cflowd auxiliary server sends an initial OSS request to retrieve the network object model from a main server, and subsequently receives JMS event notifications about updates to the objects in the model. A Cflowd auxiliary server uses FTP or SFTP to forward statistics data to an OSS or third-party application.

### 1.4.7 5620 SAM integration with external systems

The 5620 SAM can be integrated with external network management systems for purposes such as alarm forwarding. Depending on the external system type, you can use a 5620 SAM GUI contextual menu option to open a session on the external system. See the *5620 SAM Integration Guide* for information.

## 1.5 System structure

### 1.5.1 Introduction

A 5620 SAM system has a readily adaptable, modular structure that incorporates a relational data model and employs distributed processing.

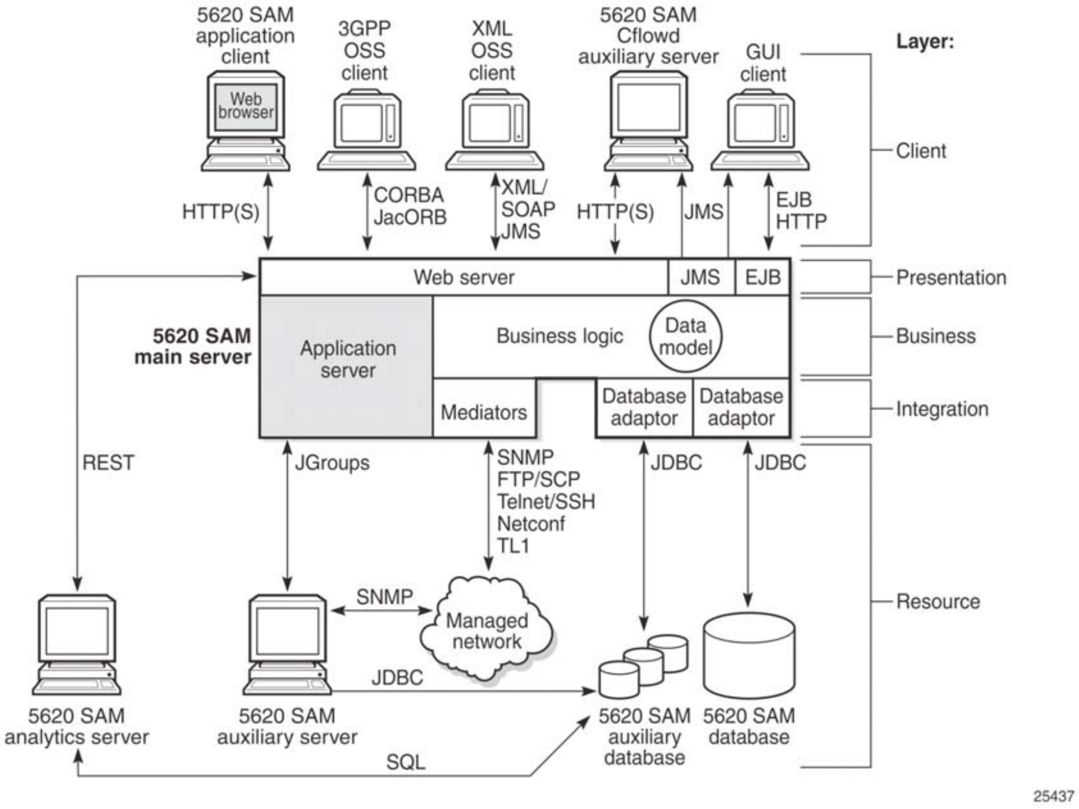
### 1.5.2 Framework

The 5620 SAM system elements are created using proprietary and third-party software, and are logically organized in a framework that has the following layers:

- resource
- integration
- business
- presentation
- client

The following figure shows the multi-layer model and the elements in each layer.

*Figure 3* 5620 SAM multi-layer model



### Resource layer

The resource layer includes the network of managed NEs, the 5620 SAM database, and optionally, one or more auxiliary servers, an auxiliary database, and one or more analytics servers. The available resources include NE configurations and software images, customer service configurations, and statistics data.

### Integration layer

The integration layer buffers resource-layer elements from the business layer. This layer contains the mediators, which communicate with equipment in the managed network, and the database adapter. The mediator components translate messages from the business layer into the SNMP, FTP, secure FTP, and CLI commands that are sent to the managed network. Messages that are received from the network are processed by the mediator components and passed to the business layer. The database adapter translates business logic requests into JDBC commands, and translates JDBC responses into Java business model objects.

#### Business layer

The business layer contains the logic and data model for 5620 SAM functions. The business logic processes client requests, SNMP traps from managed NEs, and internal server events, and performs the appropriate actions on the managed network, clients, and data model, which maintains information about network objects and their relationships. To support the business layer, an application server provides Java EE services.

#### Presentation layer

The presentation layer buffers the application logic from the client layer. This layer contains several components. The web server receives SOAP/XML messages from OSS clients and passes them to the business layer. The application server handles EJB method invocations received from the GUI clients and returns the responses generated by the business-layer logic. The application server also forwards JMS event notification messages from the business layer to GUI and OSS clients.

#### Client layer

The client layer comprises the GUI, OSS, and web-based clients. The GUI client Java VM sends EJB RMIs to a main server. The OSS clients send XML/SOAP or 3GPP CORBA messages to a main server. Web clients use JNLP for portal access.

### 1.5.3 Server data model

The server data model represents the physical and logical elements of the network, such as equipment, customers, services, and statistics. The model also describes the relationships between objects, so allows operators to perform high-level operations that are propagated to child objects, as required. The object associations enable effective central management of large, complex networks.

The 5620 SAM maintains in the data model a representation of the current managed network state, and incorporates changes as they occur. Changes that are initiated by NEs include event notifications such as fault traps and state changes; changes that are initiated by clients include object creation, deletion, and configuration updates. The changes are applied to the model, saved in the 5620 SAM database, deployed to the network as required, and reported to clients.

### 1.5.4 Distributed server architecture

The 5620 SAM server functions can be distributed across multiple physical or virtual stations in a standalone or redundant configuration.

A main server is the network management engine that monitors the managed network and processes GUI and OSS client requests. A main server also directs the operation of the associated auxiliary servers and distributes the processing load, as required. The GUI and OSS clients interact only with the currently active main server.

Auxiliary servers in a redundant 5620 SAM system respond to processing requests only from the primary main server. Depending on the system configuration, if the main servers change roles because of a failure or deliberate operator action, the active auxiliary servers begin to take requests from the new primary main server, or become idle as the main server directs the requests to other auxiliary servers.

A main server sends new or updated operating information such as the 5620 SAM license capacity, redundancy status, or database credentials, to each auxiliary server as the information becomes available.

## 1.6 Security

### 1.6.1 Introduction

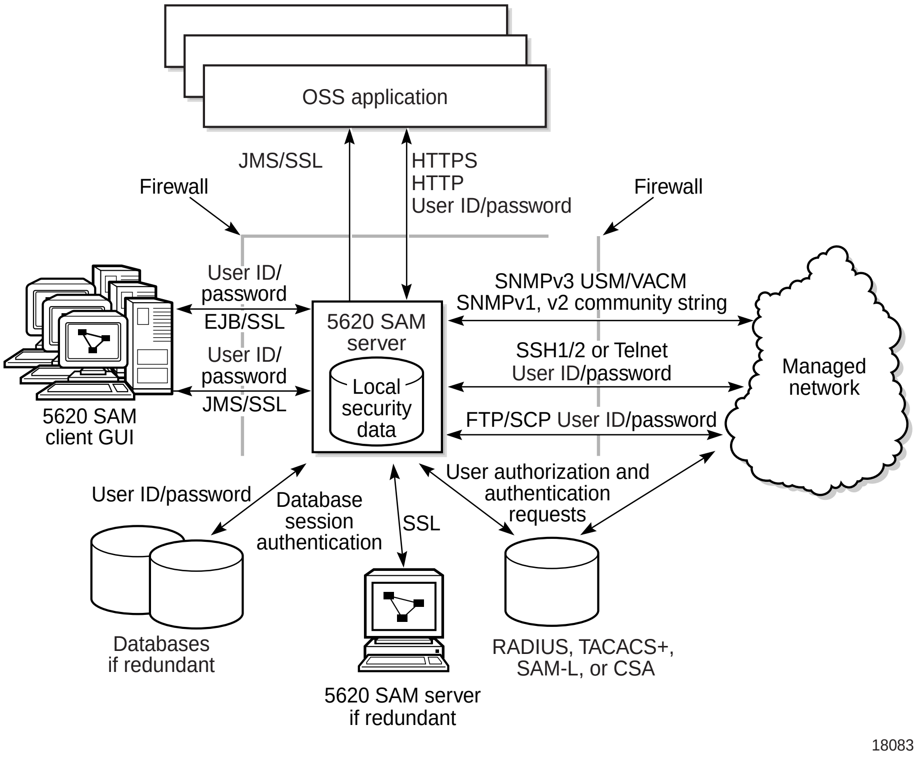
A distributed system such as the 5620 SAM requires security at the session and other communication layers. A GUI or OSS client must provide user credentials for access to the 5620 SAM.

You can protect the session credentials and messages using mechanisms and protocols that include the following:

- HTTPS, as the application-layer transport for OSS clients
- SSH, SCP, and SNMPv3 with USM or VACM, at the application layer for communication between a main server and the managed network
- SSL/TLS, at the presentation layer, between a main server and all other components, with the exception of a 5620 SAM database or auxiliary database.
- NAT, at the network layer, between the following:
  - — main server and single-user GUI client or client delegate server
  - — main or auxiliary server and OSS client
  - — main or auxiliary server and managed network
- IP validation, at the network layer, between a main server and database

The following figure shows the 5620 SAM components and the available security mechanisms.

*Figure 4* 5620 SAM security mechanisms



## 1.6.2 Session management

Effective session management requires authentication, authorization, and accounting, or AAA. Authentication is the verification of user credentials. Authorization is the assignment of access privileges to users. Accounting is the recording of user actions. A 5620 SAM operator can configure AAA functions using the local 5620 SAM security mechanisms, a third-party server, or both.

- Local 5620 SAM authentication is performed using a local database of users and a local security scheme.
- Supported third-party authentication servers are RADIUS, TACACS+, LDAP, SAM-L, and CSA, which run on separate platforms, and have separate user lists and administration processes.

5620 SAM user accounts consist of a user name, password, and an associated user group, scope of command, and span of control. User groups define user authorization levels, and control the level of access to objects such as equipment, customers, services, and alarms. A system administrator can limit the type of user access per managed NE; for example, allowing FTP access but denying console, Telnet, or SNMP access.

**Client sessions**

All client sessions require authentication.

- The client GUI EJB sessions are secured by the session username and password.
- Each OSS client message is authenticated using cached information from an authorization server.
- JMS messages are secured by the OSS client user name and password.

**Database sessions**

The 5620 SAM database is accessible through a main or auxiliary server connection that is secured by a username and password. After each database update in response to a client request, the client activity log records the request information, which includes the name of the associated 5620 SAM user.

Secure communication between a main server and a 5620 SAM database is available in the form of IP-address validation, which is configurable during 5620 SAM system installation or upgrade.

**Managed NE sessions**

A 5620 SAM server opens CLI, FTP, SFTP and SCP sessions on managed NEs. A managed NE uses a local security database, or a third-party service such as RADIUS or TACACS+, to perform AAA functions.

SNMPv3 message authentication and authorization are handled by the USM and VACM mechanisms, which define the user authorization permissions. Older SNMP versions are authenticated using community strings. Each SNMP message is individually authenticated.

### 1.6.3 Network transport security

Transport-layer security is available to the network protocols that carry messages between 5620 SAM components.

**Main server and clients**

Communication between a main server and clients is performed using XML/SOAP, EJB, or JMS messages.

- OSS clients use HTTPS to send XML/SOAP messages when SSL is enabled in a 5620 SAM system, and otherwise use HTTP.
- GUI clients use the EJB interface, which can be secured using SSL.

- JMS, which is used by GUI and OSS clients, can be secured using SSL.
- In a secured redundant deployment, the standby main server acts as an SSL client of the primary main server.

**Servers and managed NEs**

A managed NE communicates with a main or auxiliary server using SNMP, FTP, SCP, or UDP. When SNMPv3 is used, an SHA or MD5 authentication key is included in each message and checked against the shared encryption key.

SSH provides the security for a CLI session between a 5620 SAM GUI client and a managed NE.

RSA encryption is available for communication between auxiliary servers and managed NEs. Contact customer support for information.

**Firewall support**

The 5620 SAM supports firewalls on all server interfaces; for example, between a main server and the auxiliary servers and GUI or OSS clients, and between a main or auxiliary server and the managed network. See the *5620 SAM Planning Guide* for firewall and reserved TCP port information.

## 1.7 Fault tolerance

### 1.7.1 Introduction

Fault tolerance provides system reliability by maintaining availability in the event of a component failure. 5620 SAM fault tolerance includes high availability using component redundancy. Deploying redundant 5620 SAM hardware and software components ensures that there is no single point of 5620 SAM system failure.

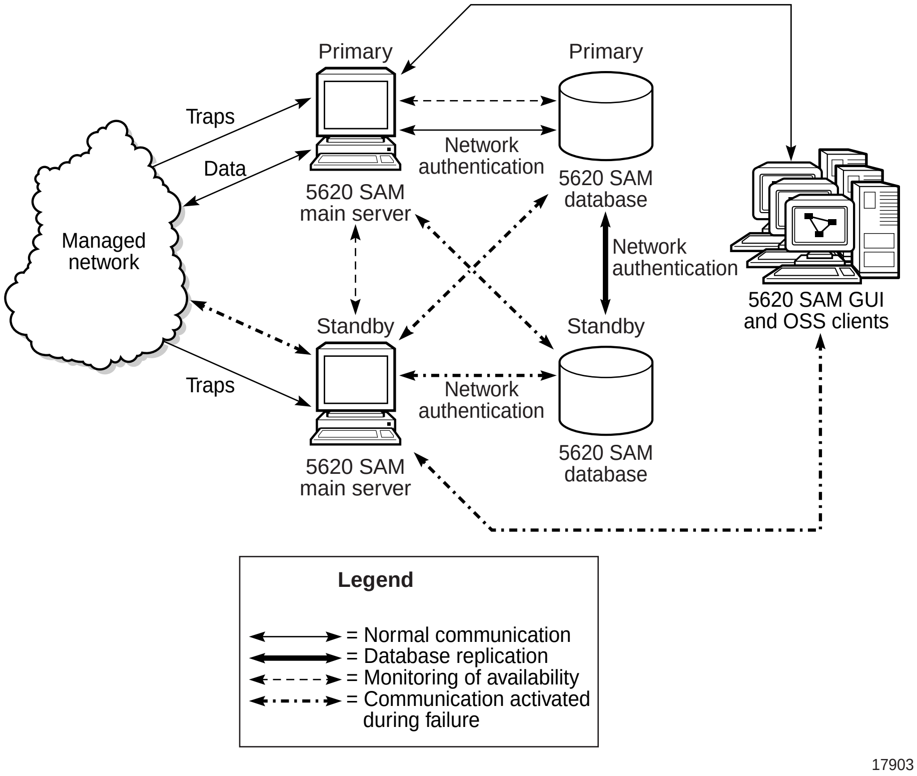
Redundant physical network interfaces and points of network entry ensure that there is no single point of failure between the 5620 SAM system and the managed network. Redundant network paths, for example, in-band and out-of-band management, can help to prevent the isolation of a main server from the network in the event of a routing failure.

See the “5620 SAM system redundancy” chapter of the *5620 SAM System Administrator Guide* for more information.

### 1.7.2 Main server and database redundancy

A redundant 5620 SAM system consists of a primary main server and an associated primary 5620 SAM database that actively manage the network, and a second server and database pair in standby mode. A 5620 SAM server and database pair can be collocated on one station or run on separate stations. The following figure shows a distributed 5620 SAM system deployed in a redundant configuration.

*Figure 5* Redundant 5620 SAM system



See the *5620 SAM System Administrator Guide* for more information about 5620 SAM redundancy.

### Main server redundancy

Main server redundancy is achieved using clustering technology provided by a JBOSS application server on each main server. The primary and standby main servers regularly poll each other to monitor availability. Traps from the managed network are always sent to both main servers in order to avoid delays in the event of a server activity switch.

If the primary server loses visibility of the standby server, it notifies the GUI clients. If the standby server loses visibility of the primary server, the standby server attempts to become the primary server by connecting to the primary database.

**5620 SAM database redundancy**

5620 SAM database redundancy uses Oracle Data Guard Replication in real-time apply mode to keep the standby database synchronized with data changes in the primary database. The supported fault-recovery operations are database switchovers and database failovers. A switchover is a manual operation that switches the primary and standby database roles. A failover is an automatic operation that forces the standby database to become the primary database when the primary database fails or becomes unavailable.

The primary main server polls each database to check the availability. If the primary or standby database is unavailable, the main server notifies the GUI clients. If both main servers lose contact with the primary database, a failover occurs and the standby database becomes the primary database.

### 1.7.3 Auxiliary servers and 5620 SAM redundancy

Auxiliary servers are passively redundant. They do not cause or initiate main server or database redundancy activities, but if a Preferred auxiliary server ceases to respond to requests from the primary main server and a Reserved auxiliary server is available, the main server directs the current and subsequent requests to the Reserved auxiliary server until the Preferred auxiliary server is again available.

An auxiliary server communicates only with the current primary server and database. After a 5620 SAM redundancy activity such as a database failover, the primary main server directs the auxiliary servers to stop communicating with the former primary component and to begin communicating with the current primary component.

### 1.7.4 Cflowd auxiliary servers and 5620 SAM redundancy

Cflowd auxiliary servers, which collect AA accounting and AA Cflowd statistics, can be configured to transfer the collected statistics files to redundant destinations. For a higher degree of fault tolerance, you can configure two or more Cflowd auxiliary servers to collect statistics from the same set of NEs and transfer the statistics files to redundant destinations. Such a configuration ensures that the statistics collection and transfer continue uninterrupted in the event that a Cflowd auxiliary server and a transfer destination fail.

### 1.7.5 Analytics servers and 5620 SAM redundancy

You can deploy multiple analytics servers in an active/active configuration that eliminates the single point of failure in the event that a server fails. The use of multiple analytics servers also allows load balancing of client requests among the servers.

A main server that provides web client access to the 5620 SAM Analytics application monitors the reachability of each analtyics server. The main server directs client requests to other analytics servers in the event that one server fails. A load-balancing scheme in the main server configuration determines how the main server distributes client requests among the available servers.

The redundancy function is active when the main server configuration includes multiple analytics servers, regardless of whether load balancing is enabled.

For additional fault tolerance, an analytics server configuration can also include the IP addresses of multiple auxiliary database stations in an auxiliary database cluster. If the auxiliary database is unreachable using one IP address, the other addresses are tried in sequence until communication with the auxiliary database is established.

## 1.8 Standards compliance

### 1.8.1 Description

The 5620 SAM system uses industry standards and open-standard interfaces that allow the system to interoperate with a variety of other network monitoring and management systems. The following table lists the 5620 SAM compliance with various standards:

*Table 1* 5620 SAM standards compliance

| Standard | Description |
|---|---|
| 3GPP | 3rd Generation Partnership Project IRPs for CORBA R8 and SOAP/XML R8 Solution Sets |
| draft-grant-tacacs-02.txt | TACACS+ client |
| draft-ylonen-ssh-protocol-00.txt | SSH |
| EJB | Java EE Enterprise Java Session Bean version 2.3 |
| HTML5 | HyperText Markup Language 5, for 5620 SAM applications |
| HTTP(S) | HyperText Transfer Protocol (Secure) version 1.1 |
| ITU-T X.721 | SMI |
| ITU-T X.734 | Event report management function |
| Java SE | Java Standard Edition version 8 |
| JBOSS EAP | Java Bean Open Source Software Enterprise Application Platform version 6 |
| JMS | Java Message Service version 1.1 |
| JSON | ECMA-404 JavaScript Object Notation Data Interchange Format |

*Table 1* 5620 SAM standards compliance (continued)

| Standard | Description |
|---|---|
| JS/ECMAScript 5 | ECMA-262 ECMA Script Language Specification |
| M.3100/3120 | Equipment and connection models |
| MTOSI | Compliance of generic network objects, inventory retrieval, and JMS over XML |
| RFC 0959 | FTP |
| RFC 1213 | SNMPv1 |
| RFC 1738 | Uniform Resource Locators (URL) |
| RFC 2138 | RADIUS client 2618 |
| RFC 3411-3415 | SNMPv3 |
| RFC 3416 | SNMPv2c |
| RFC 5246 | The Transport Layer Security (TLS) Protocol |
| RFC 6241 | Network Configuration Protocol (NETCONF) |
| SAML | SAM-L 1.1 |
| SOAP | W3C SOAP 1.2 |
| TMF 509/613 | Network connectivity model |
| TR-069 | TR-069 (Amendment 1) by way of the Home Device Manager |
| XML | W3C XML 1.0 |
| | W3C Namespaces in XML |
| | W3C XML schemas |

The following standards are considered in the 5620 SAM GUI design:

- Sun Microsystems, *Java Look and Feel Design Guidelines*, Addison-Wesley Publishing Company, Reading, Massachusetts 1999.
- ANSI T1.232-1996, *Operations, Administration, and Provisioning (OAM&P)- G Interface Specifications for Use with the Telecommunications Management Network (TMN)*.
- Telcordia (Bell Core) GR-2914-CORE Sept. 98, *Human Factors Requirements for Equipment to Improve Network Integrity.*
- Telcordia (Bell Core) GR-826-CORE, June 1994, Issue 1, Section 10.2 of OTGR, *User Interface Generic Requirements for Supporting Network Element Operations.*
- ITU-T Recommendation Z.361 (02/99), *Design guidelines for Human- Computer Interfaces (HCI) for the management of telecommunications networks.*

- ETSI EG 201 204 v1.1.1 (1997-05), *Human Factors (HF); User Interface design principles for the Telecommunications Management Network (TMN) applicable to the “G” Interface.*
- 3GPP 32-series R8 specification, published December, 2009.