NOKIA

NSP
Network Services Platform

Network Functions Manager - Packet (NFM-P)

Release 19.3

Architecture Guide

3HE-15126-AAAA-TQZZA

Issue 1

March 2019

Legal notice

Contents

About this document **4**

1 NFM-P architecture **5**

1.1 NFM-P architecture overview **5**

1.2 Network management functions **5**

1.3 System components **7**

1.4 Component communication **8**

1.5 System structure **10**

1.6 Security **13**

1.7 Fault tolerance **16**

1.8 Standards compliance **17**

About this document

Purpose

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

Document support

Customer documentation and product support URLs:

- Documentation Center
- Technical support

How to comment

Documentation feedback

1 NFM-P architecture

1.1 NFM-P architecture overview

1.1.1 Introduction

The NFM-P 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, REST, and WebDAV. The NFM-P uses open-standard interfaces that allow the system to interoperate with a variety of other network monitoring and management systems.

1.1.2 NFM-P functions

The NFM-P 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
- interworking with other network systems

1.1.3 Main architecture features

The main features of the NFM-P 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 NFM-P applications by effectively exporting XML and REST 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 NFM-P functions
- component redundancy that provides a high degree of fault tolerance

1.2 Network management functions

1.2.1 Introduction

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

An NFM-P 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.

An NFM-P 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 Documentation Center.

1.2.2 Service management

NFM-P service management allows network operators to provision customer services such as VLL, VLAN, VPLS, IES, and VPRN. Each service can be monitored to provide performance, usage, and fault information.

1.2.3 Accounting

The NFM-P collects accounting statistics from managed NEs. Depending on the deployment and required functions, the NFM-P stores, forwards, or performs post-processing on the collected statistics data. A variety of statistics reporting and presentation functions are available.

1.2.4 Equipment management

The NFM-P maintains an equipment data model and deploys configuration updates to the managed NEs. For example, when an NFM-P 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 NFM-P can monitor services and network resources using performance statistics, OAM diagnostic tools, and data validation, and raises alarms when appropriate.

- The NFM-P can collect NE performance statistics and KPI information from specified NEs.
- The NFM-P has a comprehensive suite of OAM tools for monitoring service, NE, and transport availability and performance. You can also run tests before service activation to ensure that a service functions correctly after activation.
- The NFM-P regularly compares the configuration of each managed NE with the associated information in the NFM-P database, and updates the database information accordingly.

1.2.6 Fault management

The NFM-P performs fault management in response to NE events by analyzing the events to create status updates and raise alarms, as required. GUI clients use visual and auditory cues to alert an operator to a new alarm.

The NFM-P immediately forwards fault information as JMS events to OSS clients that subscribe to the appropriate JMS topic, and in response to XML API or REST API client requests for information.

1.3 System components

1.3.1 Introduction

An NFM-P system includes several components that are described below. Some components are supported only in specific deployment types. See the *NSP NFM-P Planning Guide* for comprehensive information about the supported deployment configurations.

1.3.2 Main server

A main server is the central Java-based network-management processing engine. A main server can be collocated on one station with a main database, or installed 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 Main database

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

1.3.5 Auxiliary database

An auxiliary database is an optional, horizontally scalable database that expands the NFM-P storage capacity for demanding operations such as statistics collection.

The database is deployed on one station, or distributed among three or more stations, depending on the scale requirement. In a multi-station auxiliary database, load balancing and data replication among the stations provide high performance and robust fault tolerance.

1.3.6 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.

1.3.7 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 GUI session that is opened through a client delegate server is functionally identical to a single-user client GUI session. The client delegate server locally stores the files that are unique to each user, such as the client logs and GUI preference files.

1.3.8 OSS clients

An Online Support System, or OSS client, is an in-house or third-party application that automates GUI client tasks or retrieves data from the NFM-P. An OSS client is platform-independent, because only Java messages are exchanged with the NFM-P.

The NFM-P supports the following OSS clients:

- XML/SOAP clients—use the NFM-P XML API to perform network management; XML schema files provide the data object definitions and describe the object attributes and methods; see the *NSP NFM-P XML API Developer Guide* for information
- REST API clients—use the NFM-P REST API to perform network management; see the online REST API documentation for information

1.3.9 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/nsp/nfmp/server/nms/distribution/licenses

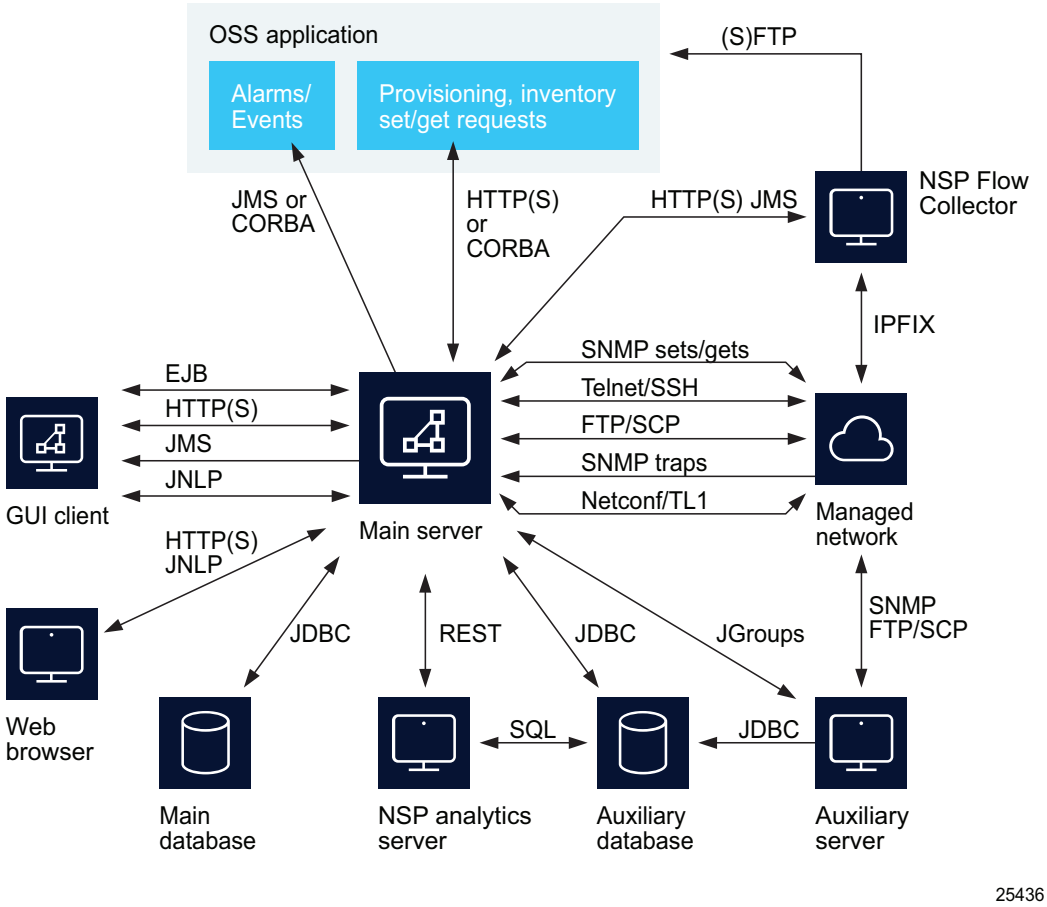
1.4 Component communication

1.4.1 Introduction

The NFM-P component interfaces use industry-standard protocols for communication among servers, databases, NEs, and clients, as shown in Figure 1-1, “NFM-P component communication” (p. 9). NFM-P components communicate with other NFM-P components and external entities using IPv4 or IPv6 exclusively, with the following exceptions:

- The NFM-P can communicate with and manage a network using IPv4 and IPv6 concurrently.
- An NFM-P GUI client or browser-based application client can connect to the NFM-P using IPv4 or IPv6, regardless of the protocol version in use between the NFM-P server and database components.

Figure 1-1 NFM-P 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 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 NFM-P of events. UDP streaming is used by NEs for operations such as forwarding PCMD records to the NFM-P.
- 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 NFM-P 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 an NFM-P 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.
- NFM-P 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 API 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 API 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 *NSP NFM-P XML API Developer Guide* for more information about the messaging between XML API clients and main servers.
- REST API OSS clients perform network management functions and receive notifications using the NFM-P REST API. See the online REST API documentation for information.

1.4.4 Main server and database

A main server communicates with a main database instance 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 NFM-P integration with external systems

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

1.5 System structure

1.5.1 Introduction

An NFM-P system has a readily adaptable, modular structure that incorporates a relational data model and employs distributed processing.

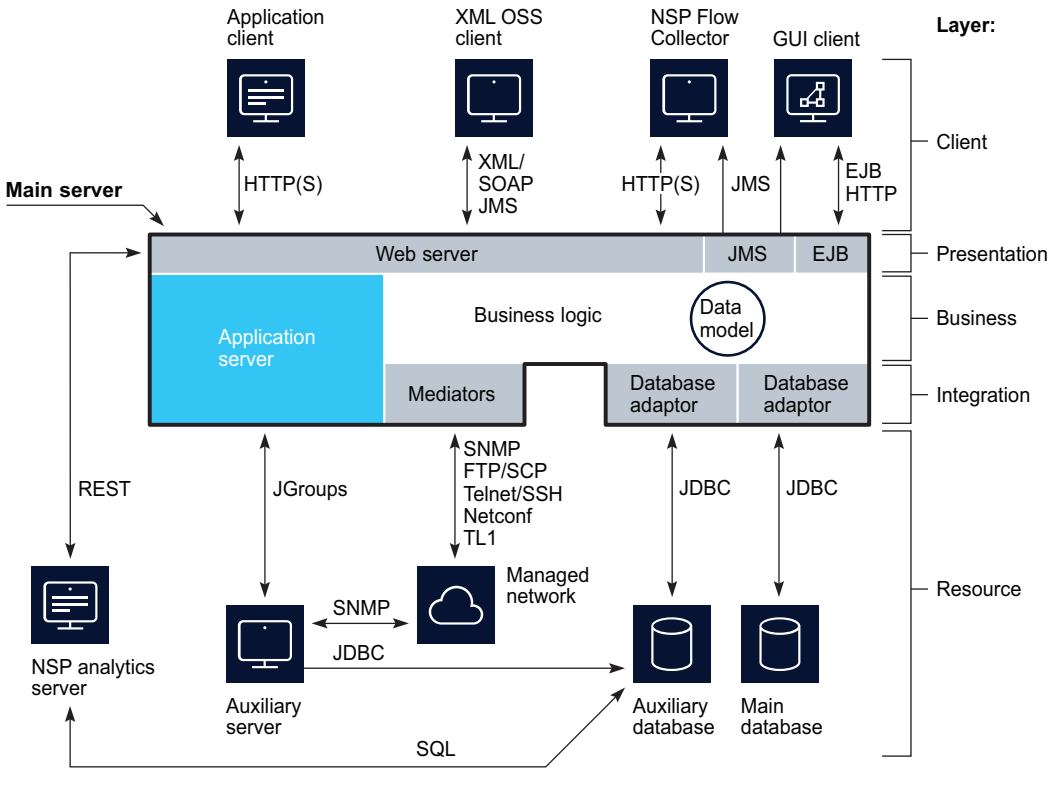
1.5.2 Framework

The NFM-P 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 1-2 NFM-P multi-layer model



Resource layer

The resource layer includes the network of managed NEs, the main database, and optional components like auxiliary servers and an auxiliary database. The available resources include, for

example, NE configuration backups and software images, ntework topology information, customer service configurations, and statistics.

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 NFM-P 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 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 application clients. The GUI client Java VM sends EJB RMIs to a main server. The OSS clients send XML/SOAP, or REST 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 NFM-P 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 NFM-P database, deployed to the network as required, and reported to clients.

1.5.4 Distributed server architecture

The NFM-P 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 components in a redundant NFM-P system respond to processing requests only from the current primary main server. Depending on the system configuration, if the main servers change roles because of a failure or deliberate operator action, an auxiliary server begins to take requests from the new primary main server, or remains idle as the main server directs the requests to other auxiliary servers.

A main server sends new or updated operating information such as the NFM-P license capacity, redundancy status, or database credentials, to each required auxiliary component as the information becomes available.

1.6 Security

1.6.1 Introduction

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

Interfaces between a main server and other system components are secured using Transport Layer Security, or TLS.

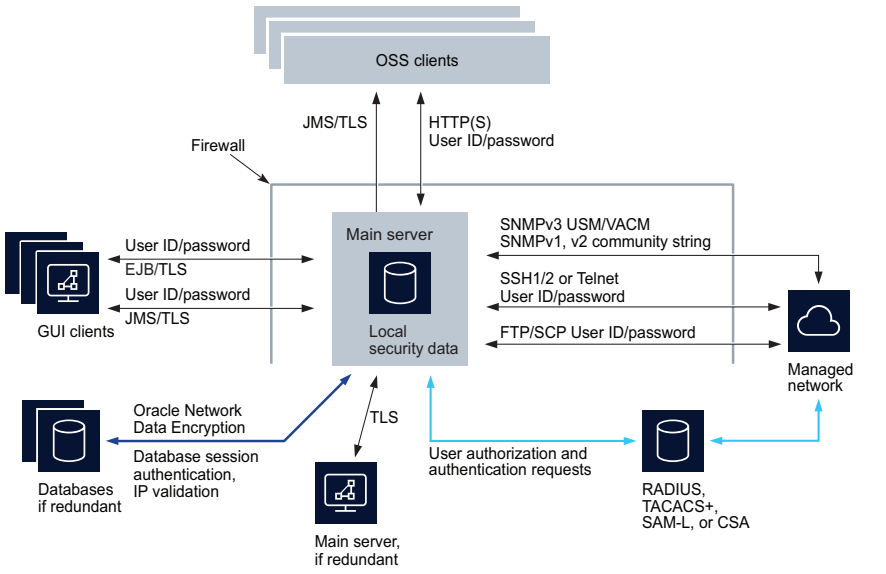
Communication with a main database is secured using Oracle Network Data Encryption.

Session credentials and messages are protected using mechanisms and protocols that include the following:

- HTTPS, as the application-layer transport for GUI and OSS clients
- SSH, SCP, and SNMPv3 with USM or VACM, at the application layer for communication between a main server and the managed network
- 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 the main or other server components and each main database

The following figure shows the NFM-P components and the security mechanisms.

Figure 1-3 NFM-P 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. An NFM-P operator can configure AAA functions using the local NFM-P security mechanisms, a third-party server, or both.

- Local NFM-P 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.

NFM-P 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. An NFM-P 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.

- A GUI client EJB session is authenticated using the client username and password.
- An OSS client session is authenticated using cached information from an authorization server.
- A JMS session is authenticated using the client username and password.

Database sessions

A main database is accessible through a connection that is secured by a user name and password. After each database update in response to a GUI or OSS client request, the client activity log records the request information, which includes the name of the associated NFM-P user.

Secure communication between a main server and database is available using the IP validation function, which is typically configured on a main database station during an installation or upgrade.

Managed NE sessions

A main or auxiliary 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 NFM-P components.

Main server and clients

Communication between a main server and clients is performed using messaging such as the following.

- XML API clients use HTTP or HTTPS to send XML/SOAP messages, and receive notifications using JMS, which can be secured using TLS.
- REST API clients use HTTPS.
- GUI clients use the EJB interface, which can be secured using TLS.

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 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 NFM-P supports firewall deployment on all server interfaces; for example, between a main server and the auxiliary servers, GUI, and OSS clients, and between a main or auxiliary server and the managed network. See the *NSP NFM-P 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. NFM-P fault tolerance includes high availability using component redundancy. Deploying redundant NFM-P hardware and software components ensures that there is no single point of NFM-P system failure.

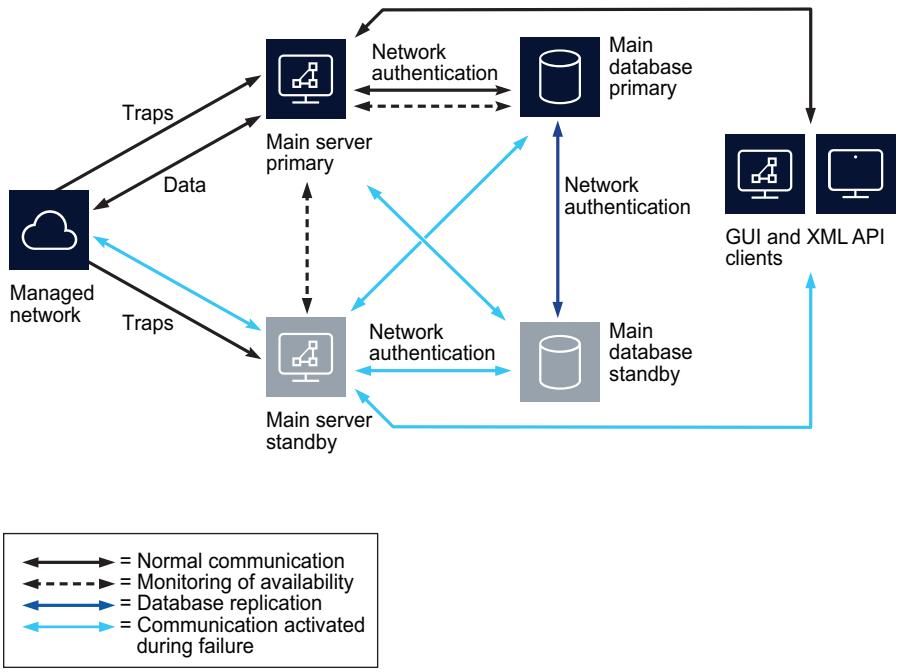
Redundant physical network interfaces and points of network entry ensure that there is no single point of failure between the NFM-P system and the managed network. Redundant network paths, for example, in-band and out-of-band management, can help to prevent the isolation of a main server from the network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

See the “NFM-P system redundancy” chapter of the *NSP NFM-P Administrator Guide* for more information.

1.7.2 Main server and database redundancy

A redundant NFM-P system consists of a primary main server and primary main database that actively manage the network, and a second main server and database in warm standby mode. A main server and database can be collocated on one station, or distributed on separate stations. The following figure shows a distributed NFM-P system deployed in a redundant configuration.

Figure 1-4 Redundant NFM-P system



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 that a main server fails.

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.

NFM-P database redundancy

NFM-P 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 a primary main database failure is detected.

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

1.7.3 Auxiliary servers and NFM-P 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 available.

An auxiliary server communicates only with the current primary server and database. After an NFM-P redundancy activity such as a database failover, the primary main server directs the auxiliary servers to communicate with the current primary component instead of the former primary component.

1.8 Standards compliance

1.8.1 Description

The NFM-P 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 NFM-P compliance with various standards:

Table 1-1 NFM-P standards compliance

Standard	Description
draft-grant-tacacs-02.txt	TACACS+ client
draft-ylonen-ssh-protocol-00.txt	SSH

Table 1-1 NFM-P standards compliance (continued)

Standard	Description
EJB	Java EE Enterprise Java Session Bean version 2.3
HTML5	HyperText Markup Language 5, for NFM-P 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
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 NFM-P 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.*