Oracle® Communications Services Gatekeeper

Concepts and Architectural Overview

Release 4.1

January 2009

Oracle Communication Services Gatekeeper Concepts and Architectural Overview, Release 4.1

Contents

1. Document Roadmap

Document Scope and Audience 1-1

Guide to This Document 1-2

Terminology 1-3

Related Documentation 1-8

2. Introducing Oracle Communications Services Gatekeeper

What Oracle Communications Services Gatekeeper Provides 2-2

Access to Telecom Network Service Capabilities Using APIs Based on Well-Known Standards 2-3

Access to Oracle Communications Converged Application Server for Connectivity to SIP Network Infrastructure 2-4

Application Development Tools 2-4

Support for Automating Partner Management Using Web Services 2-5

Common Access Control for Both Internal and Third Party Applications 2-5

Flexible Authorization Control Based on Fine-Grained Policy Decisions 2-6

Enhanced Network Protection 2-6

Built-in Network Routing 2-6

Carrier Grade and Fully Scalable Architecture 2-7

OSS and Billing System Integration 2-8

Subscriber Personalization and Protection 2-8

Extensible Architecture 2-8

3. Software Architecture Overview

Overview 3-1
Communication Services 3-2
Container Services 3-3
Deployment Model 3-6

4. Introducing Communication Services

Overview 4-1
How They Work 4-1
Typical Application-Initiated Traffic Flow 4-2
Typical Network-Triggered Traffic Flow 4-5
Platform Features 4-6
Service Level Agreements and Policy Enforcement 4-6
Service Level Agreements and Network Protection 4-8
Traffic Security 4-9
Events, Alarms, and Charging 4-9
Statistics and Transaction Units 4-10

5. Developing Applications

Overview of Interfaces 5-1
SOAP-Based Interfaces 5-1
RESTful Interfaces 5-4
Native Interfaces 5-5
References 5-6
SDK 5-6

6. Managing Application Service Providers

Overview 6-1
The Administration Model 6-2

Partner Relationship Management Interfaces 6-4
Other Tasks Associated with Administering Service Providers 6-5

7. Managing Oracle Communications Services Gatekeeper: OAM

Overview 7-1
The WebLogic and Oracle Communications Services Gatekeeper Management Console . 7-2
OAM Tasks Overview 7-2
OSS Integration 7-3

8. Charging and Billing Integration

Overview 8-1
CDR-Based Charging 8-1
Data Generation 8-2
Content-Based Charging and Accounting 8-2
Billing System Integration 8-2
Billing Gateways 8-2
CDR Database 8-3
Charging Using Diameter 8-4

9. Redundancy, Load Balancing, and High Availability

Tiering 9-1
Traffic Management Inside Oracle Communications Services Gatekeeper 9-3
Application-Initiated Traffic 9-3
Network-triggered Traffic 9-5
Registering Notifications with Network Nodes 9-8
Notification Life Span 9-8
When the Network Node Supports Primary and Secondary Notification 9-9
When the Network Node Supports Only Single Notification 9-11

Network Configuration. 9-12
Geographic Redundancy 9-14
Geo-Redundant Sites. 9-15
Applications and Geo-Redundancy. 9-16

10. Service Extensibility

Overview 10-1
The Platform Development Studio. 10-2

A. Standards and Specifications

Application-Facing Interfaces A-1
Parlay X 2.1. . . . . A-1
Parlay X 3.0. . . . . A-2
Extended Web Services. A-2
Binary SMS A-2
WAP Push A-3
Subscriber Profile LDAP A-3
RESTful APIs A-3
Native A-3
Network Protocol Plug-Ins. A-3
Security. A-8
Identity and Trust A-10

Document Roadmap

This chapter describes both the audience for and the organization of this document:

- Document Scope and Audience
- Guide to This Document
- Terminology
- Related Documentation

Document Scope and Audience

This document provides a high-level account of Oracle Communications Services Gatekeeper, its structure and its capabilities, consisting of:

- An overview of how it works and what benefits it provides
- An explanation of the interfaces it offers third-party application developers, including a description of the tools it furnishes for development and testing
- A summary view of its internals, including:
  - Service capabilities
  - Operation, administration, and maintenance (OAM) mechanisms
  - Policy enforcement
  - Security

  - Billing capabilities
  - Integration with PRM/CRMs and OSS tools
  - Product extensibility
- A description of software architecture

This document will be of use to anyone who needs a high-level understanding of how Oracle Communications Services Gatekeeper works. This includes third-party application developers who wish to integrate telephony-based functionality into their products and operator-based system developers who wish to extend the functionality of Oracle Communications Services Gatekeeper or to integrate it with PRM or OSS tools. It also includes system administrators charged with installing and maintaining Oracle Communications Services Gatekeeper. Managers, support engineers, and sales and marketing people will also find information of value here.

Guide to This Document

The document contains the following chapters:

- Chapter 1, "Document Roadmap."
- Chapter 2, "Introducing Oracle Communications Services Gatekeeper." An overview of the benefits Oracle Communications Services Gatekeeper provides application developers and network operators
- Chapter 3, "Software Architecture Overview." A high-level look at Oracle Communications Services Gatekeeper's internal architecture
- Chapter 4, "Introducing Communication Services." An overview of the communication service functionality
- Chapter 5, "Developing Applications." The interfaces offered to third-party developers and the tools available to aid in testing and development
- Chapter 6, "Managing Application Service Providers." An overview of the administration model for third-party application service providers
- Chapter 7, "Managing Oracle Communications Services Gatekeeper: OAM." Oracle Communications Services Gatekeeper's application management tool and the main operation, administration and maintenance (OAM) tasks; integrating with OSS

- Chapter 8, “Charging and Billing Integration.” Supported charging types; integrating Oracle Communications Services Gatekeeper’s internal charging mechanism with external billing and settlement systems
- Chapter 9, “Redundancy, Load Balancing, and High Availability.” Fault tolerance, high availability, and load balancing mechanisms from an application and network perspective; geo-redundancy.
- Chapter 10, “Service Extensibility.” An overview of extending Oracle Communications Services Gatekeeper by creating modules to support additional application service providers and/or network connectivity interfaces
- Appendix A, “Standards and Specifications.” Detailed description of the standards and specifications supported by Oracle Communications Services Gatekeeper’s application-facing interfaces, network-facing protocols, and security mechanisms

Terminology

The following terms and acronyms are used in this document:

- 3GPP—3rd Generation Partnership Project, a collaborative group of telecom standards bodies
- Account—A registered application or service provider. An account belongs to an account group, which is tied to a common SLA.
- Account group—Multiple registered service providers or applications that share a common SLA
- Administrative user—Someone who has privileges on the Oracle Communications Services Gatekeeper management tool. This person has an administrative user name and password.
- Alarm—The result of an unexpected event in the system, often requiring corrective action.
- API—An application programming interface
- Application—A TCP/IP based, telecom-enabled program accessed from either a telephony terminal or a computer
- Application-facing interface—The interface that Application Service Providers use to interact with Oracle Communications Services Gatekeeper

- Application Service Provider—An organization offering application services to end users through a telephony network
- AS—An application server
- Application Instance—An Application Service Provider from the perspective of internal Oracle Communications Services Gatekeeper administration. An Application Instance has a user name and a password
- CBC—Charging based on the nature of the content delivered, not on time used or simple per-use cost. Content based charging.
- CDR—Charging Data Record
- Communication Service—A facade and an enabler that together form the path through which requests travel in Oracle Communications Services Gatekeeper. Each communication service corresponds to a particular service capability.
- CORBA—Common Object Request Broker Architecture
- CPU—Central Processing Unit
- CRM—Customer Relationship Management
- DMZ—Demilitarized Zone, a physical or logical subnetwork that contains and exposes an organization's external services to a larger, untrusted network
- EAR—Enterprise Archive file
- EJB—Enterprise Java Bean
- Enabler—The Oracle Communications Services Gatekeeper layer that performs policy evaluation, routing, and protocol translation. It provides network-facing interfaces.
- End user—The ultimate consumer of the services that an application provides. An end user can be the same as the network subscriber, as in the case of a prepaid service or the end user can be a non-subscriber, as in the case of an automated mail-ordering application where the subscriber is the mail-order company and the end user is a customer to this company
- Enterprise Operator—See Application Service Provider
- Enterprise Service Bus—A middleware component that supports messaging, routing, XML data transformation, and service orchestration
- ETSI—The European Telecommications Standards Institute, a telecom standards body

- Event—A traceable, expected occurrence in the system, of interest to the operator
- EDR—Event Data Record
- EWS—Extended Web Services, a set of Web Services interfaces developed by Oracle offering access to network functionality not covered by Parlay X.
- Facade—A set of interfaces exposed to application service developers. A facade functions as a view of an enabler.
- HA—Mechanisms set up to insure high availability
- HTML—Hypertext Markup Language
- HTTP—Hypertext Transfer Protocol
- INAP—Intelligent Network Application Part, a telephony signalling protocol
- Interceptor Stack—A flexible set of chained evaluation steps used in Oracle Communications Services Gatekeeper
- IP—Internet Protocol
- JDBC—Java Database Connectivity, the Java API for database access
- JEE—Java Enterprise Edition
- JMS—Java Message Service
- JMX—Java Management Extensions
- LDAP—Lightweight Directory Access Protocol
- Location Uncertainty Shape—A geometric shape surrounding a base point specified in terms of latitude and longitude. It is used in terminal location.
- MAP—Mobile Application Part
- Marshall— Record the state and codebase(s) of an object in such a way that when the marshalled object is "unmarshalled," a copy of the original object is obtained, possibly by automatically loading the class definitions of the object.
- Mated pair—Two physically distributed installations of Oracle Communications Services Gatekeeper nodes sharing a subset of data allowing for high availability between the nodes
- MIB—Management Information Base

- MLP—Mobile Location Protocol
- MM7—A multimedia messaging protocol specified by 3GPP
- MMS—Multimedia Message Service or an instance of this service
- MMSC—Multimedia Message Service Center
- Network plug-in—The Oracle Communications Services Gatekeeper module that implements the interface to a network node or OSA/Parlay SCS through a specific protocol
- NS—Network Simulator
- OAM—Operation, Administration, and Maintenance
- OASIS—The Organization for the Advancement of Structured Information Standards, an e-business and web standards consortium
- OCSG—Oracle Communications Services Gatekeeper
- Operator—The party that manages Oracle Communications Services Gatekeeper. Usually the network operator
- On-boarding—Registering applications and service providers to enable their access to Oracle Communications Services Gatekeeper and the underlying network
- ORB—Object request broker
- OSA/Parlay—The Open Service Access interfaces used by a Parlay gateways
- OSS—Operation Support Systems
- Out of the box—The level of functionality available in the default installation of Oracle Communications Services Gatekeeper
- PAP—Push Access Protocol
- Parlay—The Parlay Group, a telecom standards body
- Parlay Gateway—A telecom gateway implementing Parlay interfaces
- Parlay X—A set of telecom Web Services interfaces specified by the Parlay Group
- Plug-in—See Network Plug-in
- Plug-in Manager—The Oracle Communications Services Gatekeeper module charged with routing an application-initiated request to the appropriate network plug-in

- POJO—Plain Old Java Object
- Presence information—A status indicator that conveys the accessibility and the willingness of a potential communication partner
- Presentity—A supplier of presence information.
- PRM—Partner Relationship Management
- Quotas—An access rule based on an aggregated number of invocations. See also Rates
- RAM—Random Access Memory
- RAID—Redundant Array of Independent Disks
- Rates—An access rule based on allowable invocations per time period. See also Quotas
- RESTful—Interfaces that follow Representation State Transfer style
- Rf—The Diameter offline charging mode
- RMI—Remote Method Invocation
- Ro—The Diameter online charging mode
- SAML—Security Assertion Markup Language
- SCF—Service Capability Function or Service Control Function, in the OSA/Parlay sense.
- SCS—Service Capability Server, in the OSA/Parlay sense. Oracle Communications Services Gatekeeper can interact with these on its network-facing side
- Service Capability—Support for a specific kind of traffic within Oracle Communications Services Gatekeeper. Defined in terms of communication services
- SIP—Session Initiation Protocol
- SLA—A service level agreement
- SMPP—Short Message Peer-to-Peer Protocol
- SMS—Short Message Service, or an instance of this service
- SMSC—Short Message Service Center
- SNMP—Simple Network Management Protocol
- SOA—Service Oriented Architecture

- SOAP—A protocol for exchanging Web Services messages
- SPI—Service Provider Interface
- SQL—Structured Query Language
- SS7—Signalling System #7, a signaling protocol used in traditional telecom networks
- Subscriber—A person or organization that signs up for access to an application. The subscriber is charged for the application service usage. See End user
- TCP—Transmission Control Protocol
- TUPS—Transaction Units Per Second
- URI—Uniform Resource Identifier
- URL—Uniform Resource Locator
- USSD—Unstructured Supplementary Service Data
- VAS—Value Added Service
- VASP—Value Added Service Provider
- VLAN—Virtual Local Area Network
- VPN—Virtual Private Network
- W3C—The World Wide Web Consortium, a web standards group
- WAP Push—A protocol for sending WAP content (an encoded message including a link to a WAP address) that is pushed to a subscriber’s handset
- Watcher—A consumer of presence information
- WS-Security—An OASIS security standard for Web Services
- WSDL —Web Services Definition Language
- XML—Extensible Markup Language

Related Documentation

This architectural overview is a part of the Oracle Communications Services Gatekeeper documentation set. The other documents include:

- *System Administrator's Guide*
- *Integration Guidelines for Partner Relationship Management*
- *SDK User Guide*
- *Managing Accounts and SLAs*
- *Statement of Compliance and Protocol Mapping*
- *Application Development Guide*
- *Communications Services Reference*
- *Handling Alarms*
- *Licensing*
- *Installation Guide*
- *Platform Development Studio - Developer's Guide*
- *RESTful Application Development Guide*

Additionally, many documents in the Oracle WebLogic Server documentation set are of interest to users of Oracle Communications Services Gatekeeper, including:

- *Introduction to Oracle WebLogic Server* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/intro/
- *Installation Guide* at http://download.oracle.com/docs/cd/E12840_01/common/docs103/install/index.html
- *Managing Server Startup and Shutdown* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/server_start/
- *Getting Started With WebLogic Web Services Using JAX-WS* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/webserv/index.html
- Developing Manageable Applications with JMX at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/jmxinst/
- *Configuring and Using the WebLogic Diagnostics Framework* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/wldf_configuring/
- *Using Clusters* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/cluster/

- *Securing WebLogic Server* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/secmanage/

Introducing Oracle Communications Services Gatekeeper

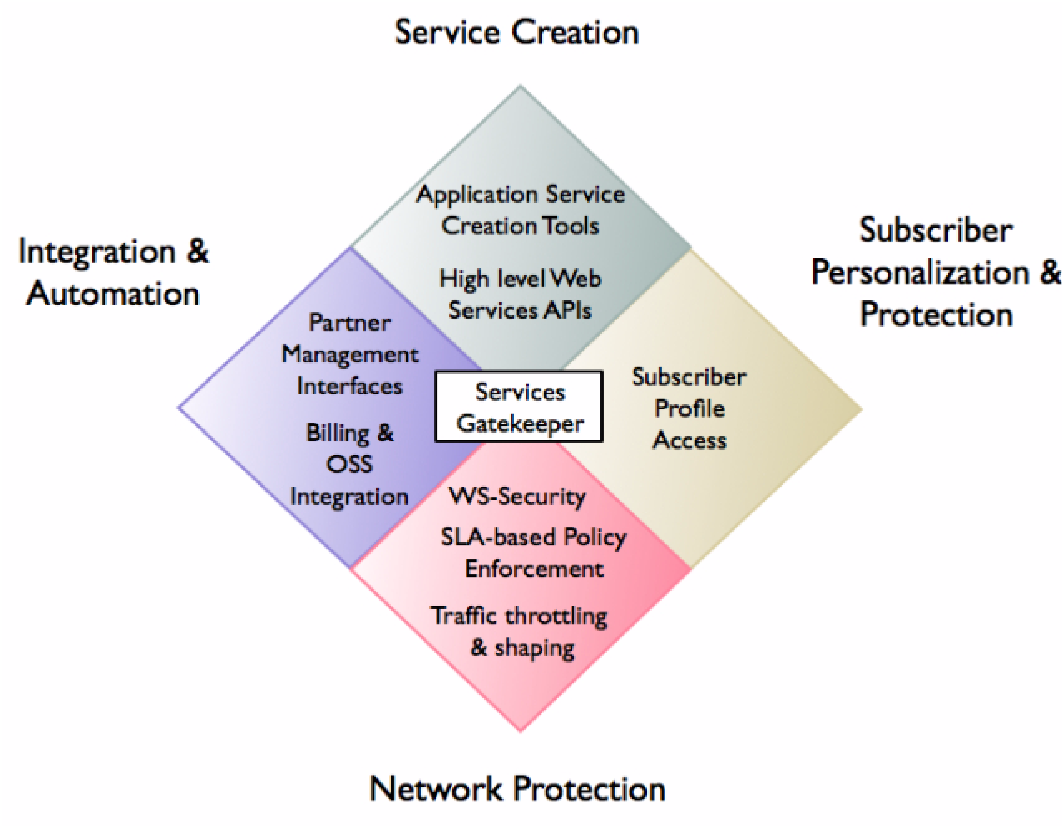
The worlds of TCP/IP applications and of telephony networks continue to converge. Subscribers want services that provide them with functionality and flexibility that cross the traditional boundaries between the world of the Internet and the world of their phones. Operators want to be responsive to those desires, and to provide services that will satisfy subscriber demands, promote subscriber loyalty, increase average revenue per user (ARPU), and drive traffic to their networks.

But developing these services has historically been complex and ungainly. What is needed is a way to reduce the overhead of creating the applications that provide those services and to enable a wider ranging development community to contribute. To do this, operators need to have a way to:

- Offer simplified access to their network's capabilities, for both internal developers and external partners
- Provide tooling and support for application service development and testing
- Manage external partners efficiently
- Protect the security and stability of the underlying network
- Integrate new services with their existing operations and management facilities
- Protect subscriber privacy and control
- Support flexibility of access as networks change and grow
- Do all this in a way that scales and meets the performance needs subscribers have come to expect

Oracle Communications Services Gatekeeper has been created specifically to help operators meet these challenges.

Figure 2-1 Oracle Communications Services Gatekeeper in Context



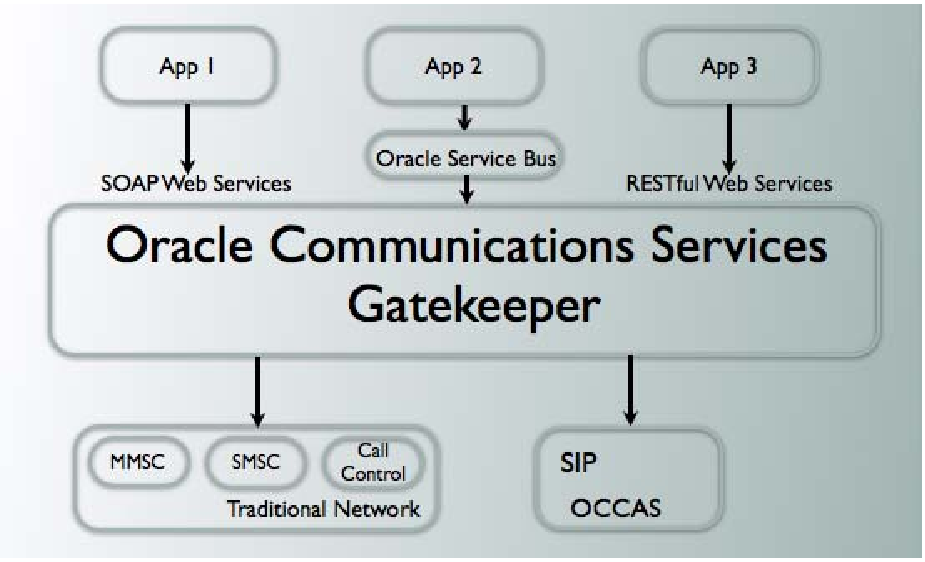
What Oracle Communications Services Gatekeeper Provides

Oracle Communications Services Gatekeeper, built using a version of Oracle WebLogic Server 10.3 (http://download.oracle.com/docs/cd/E12840_01/wls/docs103/sitemap.html) that has been hardened and extended to support the specialized needs of telecom networks, offers a host of benefits for both application service developers and operators.

Access to Telecom Network Service Capabilities Using APIs Based on Well-Known Standards

The protocols required by underlying telecom network capabilities are often complex, and the learning curve associated with achieving competence in using them is steep. To lower the barriers to entry for application service developers, out of the box Oracle Communications Services Gatekeeper provides access to standard network capabilities such as SMS, MMS or Call Control through a set of easy-to-use interfaces (called *facades* in Oracle Communications Services Gatekeeper). Depending on the desires of the operator, Oracle Communications Services Gatekeeper can offer SOAP style facades based on well-known standards such as Parlay X 2.1 and 3.0, RESTful style facades, and, in some cases, native protocol interfaces. SOAP style interfaces are also pre-integrated into an Oracle Service Bus environment, which offers application developers SOAP-based interfaces while allowing the operator the flexibility of SOA. In cases where access to desired functionality (WAP Push, Binary SMS, and Subscriber Profile) has not yet been incorporated into standardized forms, Oracle has created Extended Web Services interfaces. SOAP Web Services interfaces are published as standard WSDL files, so application service developers can use their choice of toolsets, while RESTful Web Services interfaces are designed for ease of use in pure HTTP environments. Developers can focus on creating compelling and innovative services, leaving the Communication Services components of Oracle Communications Services Gatekeeper to handle the mechanics of interacting with the various underlying network elements.

Figure 2-2 Standardized Application Interfaces



Access to Oracle Communications Converged Application Server for Connectivity to SIP Network Infrastructure

In addition to providing access to traditional telecom network functionality, Oracle Communications Services Gatekeeper can also connect application services to SIP-based functionality, using Oracle Communications Converged Application Server. Calls set up using the Parlay X 2.1 or RESTful Third Party Call communication services can be routed through SIP. Parlay X 2.1 or RESTful Call Notifications can be established using SIP and Parlay X 2.1 or RESTful Presence *watchers* (consumers of presence information) and *presentities* (providers of presence information) can be set up.

Application Development Tools

Oracle Communications Services Gatekeeper always provides:

- Web Services WSDL files
- Application Development Guide
- RESTful Application Development Guide

To further assist application service developers, Oracle Communications Services Gatekeeper can optionally provide:

- The Oracle Communications Services Gatekeeper Simulator, which supports early application development cycles using SOAP-based interfaces without requiring the developer to run a full-fledged copy of Oracle Communications Services Gatekeeper
- The Oracle Communications Services Gatekeeper Simulator GUI, a graphical test and verification environment for SMS, MMS, Terminal Location, and WAP Push traffic that runs on top of the Oracle Communications Services Gatekeeper Simulator.
- A set of Oracle WorkShop for WebLogic controls

Support for Automating Partner Management Using Web Services

Managing a large number of services, particularly when the providers are third-party partners, can be time and effort intensive. As the market expands, with niche players and short-term services being added to the more mainstream mix, the logistics of on-boarding can become very complex. To assist operators in handling processes such as partner registration, service activation and provisioning, Oracle Communications Services Gatekeeper can supply its Partner Relationship Management interfaces. These Web Services interfaces can be used to support the automating of a wide range of partner-related tasks and to provide partners with easily available access to information about their accounts. The interfaces also allow operators to create groups of partners sharing sets of data, which can be used for tiering or segmentation of partners. Operators can then focus their administrative and partner management resources on their most rewarding partners.

Common Access Control for Both Internal and Third Party Applications

Oracle Communications Services Gatekeeper can function as a single point of contact for access to the functionality of the underlying network, providing common authentication, authorization, and access control procedures for all applications, both internal and third-party based. For SOAP-based interfaces, Oracle Communications Services Gatekeeper leverages the flexible security framework of Oracle Web Logic Server 10.3 to provide robust system protection. Applications can be authenticated using plaintext or digest passwords, X.509 certificates, or SAML 1.0/1.1 tokens. Service requests can use XML encryption, based on the W3C standard, for either the whole request message or specific parts of it. And, to ensure message integrity, requests can be digitally signed, using the W3C XML digital signature standard. For RESTful interfaces, Oracle Communications Services Gatekeeper uses HTTP basic authentication: username/password and SSL.

Flexible Authorization Control Based on Fine-Grained Policy Decisions

Oracle Communications Services Gatekeeper's powerful and responsive policy enforcement mechanism uses service level agreements (SLAs) to regulate service provider and application access to particular communication service functionality down to the level of supported operations and parameters. It also supports a range of quality-of-service guarantees that can be modulated by Time of Day/Day of Week, Rates, and Quotas. If desired, further rules covering access can also be added. And service provider and application accounts can be divided into groups to simplify SLA management and maintenance. Although Oracle Communications Services Gatekeeper ships with a set of broadly comprehensive SLAs, custom versions can also be created.

In addition, subscriber permissions and preferences can be reflected in a separate Subscriber SLA, created by the operator or an integrator using tools available in the Platform Development Studio. Subscribers can indicate, for example, that they wish to allow Service Provider X to query for the location of their mobile terminals, but not Service Provider Y.

Enhanced Network Protection

In addition to the service level agreements that cover access to functionality within Oracle Communications Services Gatekeeper itself, further SLAs explicitly define service provider access to underlying network nodes. In conditions of heavy load Oracle Communications Services Gatekeeper employs throttling and shaping to protect the underlying network, prioritizing traffic based on these Node SLAs.

Built-in Network Routing

Oracle Communications Services Gatekeeper provides an internal system for the routing of service requests directly to appropriate network nodes, based on a variety of parameters, including sending application, destination address, or any arbitrary request parameter. Oracle Communications Services Gatekeeper supports in-production deployment of multiple instances of most network protocol plug-ins (the module that interacts most directly with the underlying nodes) on an as needed basis; as a result routing can be managed in a very fine-grained and powerful way.

Carrier Grade and Fully Scalable Architecture

Based on Oracle WebLogic Server 10.3's rock solid performance and superior clustering support, Oracle Communications Services Gatekeeper's architecture is designed to support the rigorous demands of telecom operators:

- Tiering:

  Oracle Communications Services Gatekeeper is deployed in two tiers, which can be separated by a firewall for increased security. State is held only in the network-facing tier, and each tier can be built out independently of the other.

- High availability and failover

  Oracle Communications Services Gatekeeper is designed throughout to ensure multi-level protection against single points of failure.

- Geo-redundancy

  To protect the system in the face of catastrophic failure, geographically distant sites can be set up as site pairs. Service Provider and Application Group SLA enforcement is synchronized across geographic sites and SLAs are enforced between the site pairs. Any changes in account configuration information are also replicated across sites.

- Storage Service

  All traffic that passes through Oracle Communications Services Gatekeeper is transactionally wrapped. Maintaining state consistently and durably in clustered and high performance environments is traditionally difficult, but Oracle Communications Services Gatekeeper's Storage Service uses a sophisticated strategy of optimizing storage based on state access patterns. An in-memory store distributed among all the nodes serves as the entrance to data access. Reading from disk, and its attendant overhead, is reduced because the disk-based database functions as an archive rather than as a system of first use. This has two important benefits:

  - Speed: Because the data is available in memory, access is extremely rapid.
  - Scalability: As a system scales out, relying exclusively on disk-based database access often becomes a performance bottleneck. Because the data in Oracle Communications Services Gatekeeper is distributed among the network tier nodes, adding additional servers to the network tier actually increases data availability.

  In addition, the Storage Service optimizes access to exactly the kinds of data that matter most in telecom traffic processing. Designed as a POJO java.util.Map-based API, client access is simplified for both storing data and making retrieval queries.

OSS and Billing System Integration

All or selected parts of the Oracle Communications Services Gatekeeper management mechanism can be integrated with an operator's external Operation Support Systems through JMX/JMS or SNMP interfaces. The tasks associated with administering current service providers and adding new ones can be simply folded into existing systems.

Oracle Communications Services Gatekeeper's internal charging mechanisms can also be integrated with an operator's existing billing systems. Offline and online (using the Parlay X 3.0 Payment API) Diameter-based charging is supported.

Subscriber Personalization and Protection

Using Oracle Communications Services Gatekeeper, applications can customize their offerings by accessing subscriber profile information stored on network LDAP servers. At the same time, operators can protect subscriber privacy by using filters based on those same profiles to regulate the access that applications have, limiting the information that applications can acquire to what the subscriber wants to make available.

In addition, if they choose, operators can define a Subscriber SLA, which creates service provider groupings called service classes that can be associated with individual subscriber URIs. The mechanism to do this is created by the operator or integrator using the Profile Provider SPI provided as part of the Platform Development Studio. The use of a Subscriber SLA allows subscribers to customize their interactions with application service providers while keeping all their subscriber data within the confines of the operator's domain.

Extensible Architecture

A flexible architecture using the robust capabilities of Oracle WebLogic Server 10.3 means that operators can extend existing communication services to support new network interfaces, for example, Unstructured Supplementary Service Data. They can also create entirely new communication services to allow application service developers access to their network's unique features, using Oracle Communications Services Gatekeeper's Platform Development Studio.

CHAPTER 3

Software Architecture Overview

The following chapter provides an overview of Oracle Communications Services Gatekeeper's software architecture, including:

- Overview
- Communication Services
- Container Services
- Deployment Model

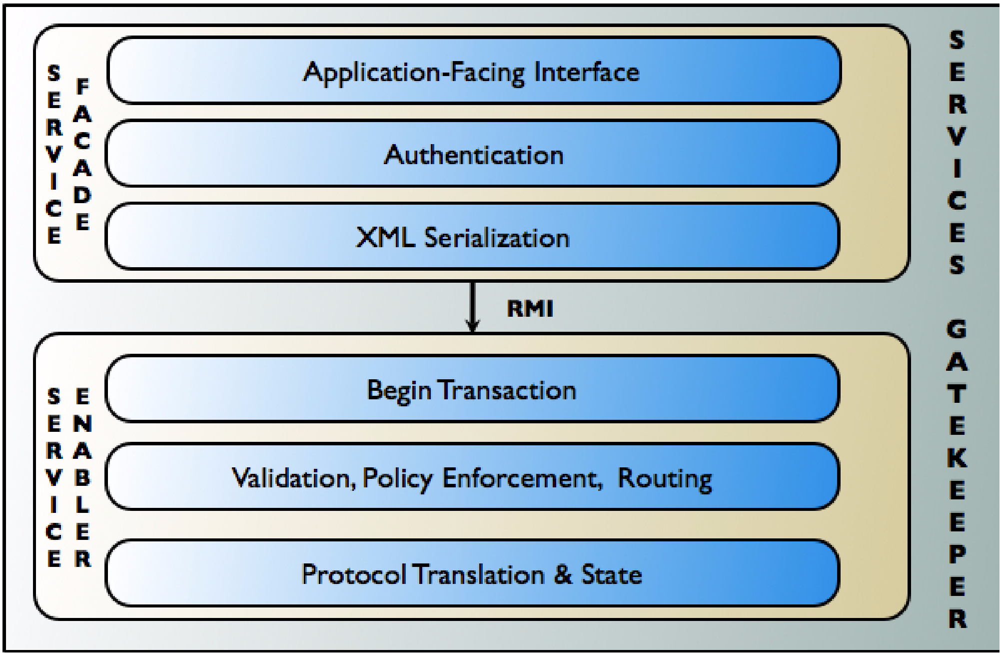
Overview

Oracle Communications Services Gatekeeper provides a robust, secure and highly performant container optimized for the task of running *communication services*. Communication services are specialized components that allow telephony network operators to provide Internet-based application developers with a powerful but simple way to access the operator's network services. Out of the box, Oracle Communications Services Gatekeeper supplies communication services providing access to such network capabilities as messaging, call control, terminal location, and presence. Extending the provided communication services or creating entirely new ones, based on the specific needs of the operator's circumstances, is made easy with the supplied Platform Development Studio. Built on Oracle WebLogic Server 10.3, Oracle Communications Services Gatekeeper is closely aligned with JEE standards and tightly integrated with Oracle Communications Converged Application Server.

Communication Services

Communication services are components that run in the Oracle Communications Services Gatekeeper container. They provide the main functionality of Oracle Communications Services Gatekeeper, exposing network capabilities to Internet based applications in a form that is easy to use and manage. All traffic in Oracle Communications Services Gatekeeper is processed through these services. A communication service consists of a *service facade*, consisting of an application-facing interface which communicates with the application and a security layer, which handles authentication and a *service enabler*, consisting of a processing layer, where requests are validated, evaluated according to service level agreements (SLAs), and routed, and a protocol translation layer, which communicates with the underlying network element. A more detailed description can be found in Chapter 4, "Introducing Communication Services."

Figure 3-1 Communication service structure



Container Services

Oracle Communications Services Gatekeeper provides a container that is highly optimized for running communication services, the Oracle Communications Services Gatekeeper components that provide the interface between Internet-based applications and the functionality of the underlying telephony network. The container leverages the many standard container services that Oracle WebLogic Server 10.3 provides, but adds a number of services designed for the specialized needs of communication services and Oracle Communications Services Gatekeeper generally. See Figure 3-2 and Figure 3-3 below for some typical uses of these services. They include:

- Budget

  Manages cross-cluster bandwidth allocation, and supports geo-redundant installations. In the context of quota and rate SLAs, it also maintains an historical perspective on usage patterns.

- Policy

  Wraps the policy engine from older versions, which can still be used

- EDR

  Broadcasts events and manages their translation into charging data and alarms, as necessary

- Storage

  Provides transparent access to data storage using distributed caching and the database

- Core

  Performs initial setup tasks

- Event Channel

  Broadcasts events among modules and servers in the cluster

- Configuration

  Stores largely read-only data, such as configuration information

- CORBA/Orbacus

  Initializes the Orbacus ORB and makes it the default for the system. Can be disabled for systems that do not connect to Parlay Gateways

- Statistics

  Generates system statistics

- Geo-Redundancy

  Provides support for geo-redundant installations

- Plug-in Manager

  Manages the processing layer

- SNMP

  Provides SNMP service for alarms

- Account

  Manages Service Level Agreements and sessions.

The examples below show interactions between the Parlay X 2.1 Short Messaging to SMPP communication service and selected container services.

Figure 3-2 Container Services in Typical Application-Initiated Traffic

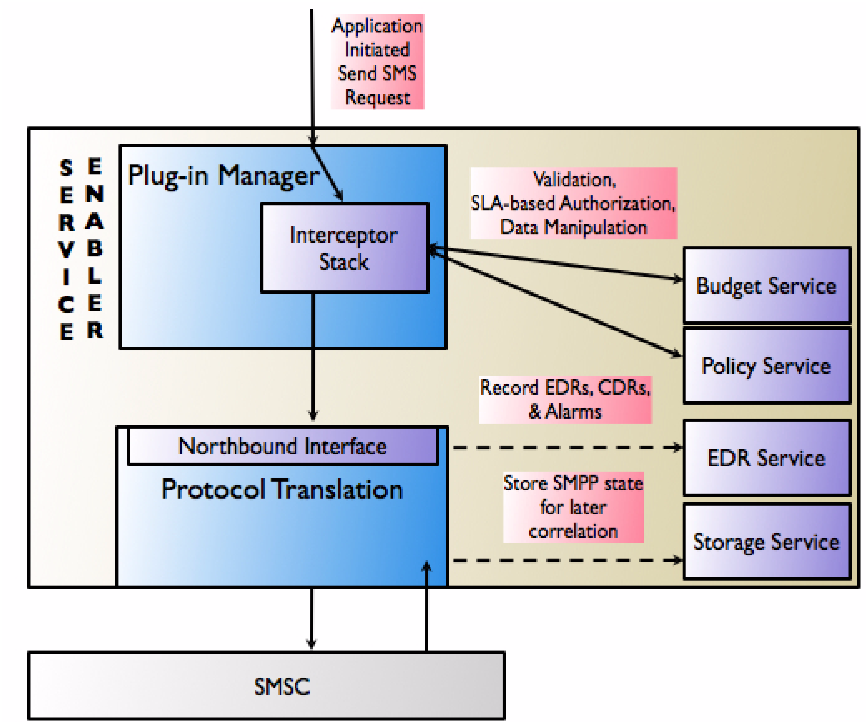
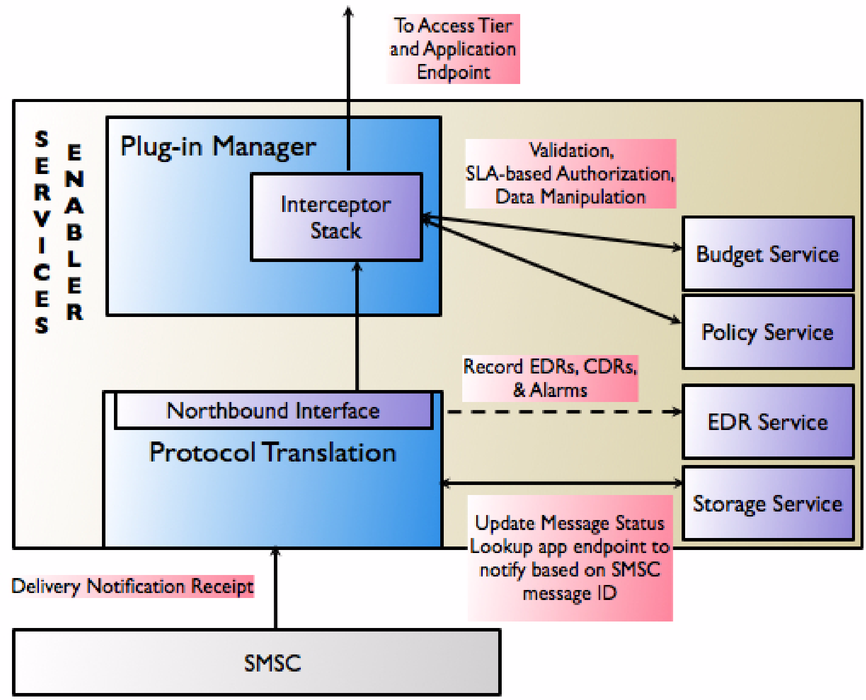


Figure 3-3 Container Services In Typical Network Triggered Traffic



Deployment Model

In production mode, communication services are typically deployed in two clustered tiers, an Access Tier and a Network Tier, separated, if desired, by a firewall. In a single physical site installation, this corresponds to a single WebLogic Server administration domain. Each communication service is deployed in its own EAR file, one per tier.

Note: Some EARs may contain either multiple application-facing interfaces (Parlay X 2.1 Short Messaging and Binary SMS/SMPP) or multiple network plug-ins (Parlay X 2.1 Third Party Call/SIP and INAP) that support the same basic service capability.

Single communication services can be installed or removed without having an impact on other communication services. If no interfaces are changed, existing communication services can be upgraded while traffic is running. This process is called a *hitless upgrade* and tracks traffic so that in-flight requests can be completed before the older version is undeployed. Communication services may be deployed selectively, as needed.

Introducing Communication Services

The following chapter presents a high level introduction to Oracle Communications Services Gatekeeper's communication services, including:

- Overview
- Platform Features

A separate document, *Communication Services Reference*, offers a more detailed look at specific services.

Overview

All application service request data flows through Oracle Communications Services Gatekeeper using communication services. A communication service consists of a service type (Multimedia Messaging, Terminal Location, etc.), an application-facing interface (also called a "north" interface), and a network-facing interface (also called "south" interface).

How They Work

Communication services are separated into two functional layers: the service facade and the service enabler. The service facade contains the application-facing interfaces, and manages interactions with applications. The service enabler contains the mechanisms necessary for dealing with the underlying network nodes.

Application-initiated requests (also called mobile terminated, or MT) enter through the service facade. A facade comprises a set of application-facing interfaces of a particular type. Oracle

Communications Services Gatekeeper supplies facades for traditional SOAP Web Services interfaces, RESTful interfaces, and, in two cases, MM7 and SMPP, native telephony interfaces. There is also a facade specifically designed to work with the Oracle Service Bus, for SOA-style installations.

After the requests have been processed by the service facade, they are sent on to the service enabler using RMI. The service enabler layer manages service authorization and policy enforcement, charging, and traffic throttling and shaping. Then the enabler translates the request into a form appropriate for the underlying network node.

Note: Although the operator may choose instead to run in a sessionless mode, by default Oracle Communications Services Gatekeeper requires that applications (except those using native telephony interfaces) acquire an Oracle Communications Services Gatekeeper session before beginning to send request traffic. Applications do this using the Session Manager interface appropriate for their facade type, which returns a session ID. The application then adds this session ID to the header of all its requests. Oracle Communications Services Gatekeeper can use this value to keep track of all the traffic that an application sends for the duration of the session.

Network-triggered (also called mobile originated, or MO) traffic is also supported by Oracle Communications Services Gatekeeper, enabling applications to receive data from the telecom network. To do so, the application must first send a request to Gatekeeper (or have the operator perform the equivalent task using OAM methods) to register a description of the types of data it is interested in - delivery notifications, incoming messages, etc. - and any criteria that the data must be meet to be acceptable. For example, an application might specify that it is only interested in receiving incoming SMSes that are addressed to the *short code* “12345” and that begin with the string “blue”.

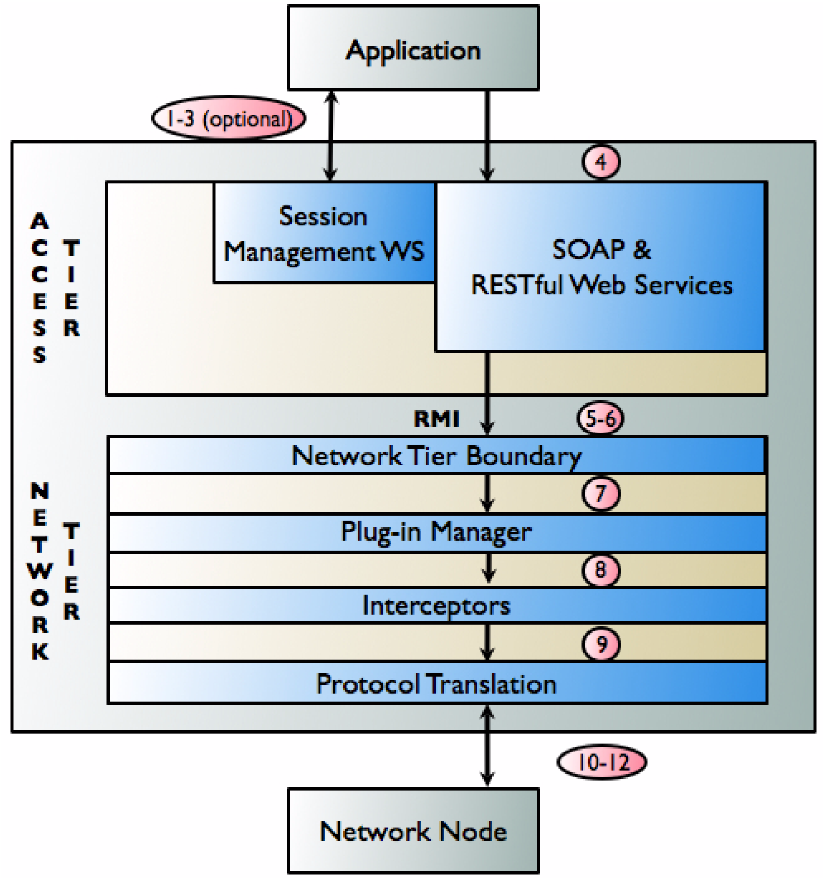
Note: For more on short codes, see “Parlay X 2.1 Short Messaging Communication Service” in *Communication Services Reference*, another document in this set.

Typical Application-Initiated Traffic Flow

Figure 4-1 illustrates typical application-initiated traffic flow.

Note: The two native interfaces, MM7 and SMPP, behave in slightly different ways.

Figure 4-1 Typical Application-Initiated Traffic Flow



1. *Steps 1-3 are optional and may be turned off.* An application establishes a session by using the Session Management Web Service in the Facade layer.
2. A session is established, and the SessionID is returned to the application. Once the application has been established, it may access multiple communication services across the cluster transparently.
3. The session is valid until the application terminates it or an operator-established time period has elapsed.

   Note: Sessions allow correlation among sequences of operations. They are not used for purposes of authentication.

4. A request for a particular operation, usually transported over SSL, enters at the application-facing interface in the facade layer, either directly from the application, or, if the particular installation uses an Oracle Service Bus (OSB), from the OSB. This interface is implemented as a SOAP- based Web Service or a RESTful Web Service. Requests using the RESTful requests are authenticated with HTTP basic authentication using username/password. SOAP-based Web Services requests are authenticated using WebLogic Server's WS-Security, which supports plaintext or digest passwords, X.509 certificates, or SAML tokens.

   Note: All requests are authenticated in this manner, whether the application uses the session mode or not.

   In addition, SOAP-based requests may be further secured through encryption using the W3C's standard XML encryption and through digital signatures using the W3C XML digital signature standard. The particular security requirements of the installation are specified in the WS-Policy section of the operator published WSDL file.

   Note: It is possible to use the appropriate standard Parlay X 2.1 or 3.0 WSDL to create SOAP-based requests, but the developer would then be required to ascertain the appropriate security type from the operator and insert the information manually.

5. The request is serialized and is passed on to the service enabler over RMI.

   Note: From this point on in the flow, requests that enter the communication service using the SOAP Service Facade and those using the RESTful Service Facade use the same service enablers. SLA construction, CDRs/EDRs/Alarms, and so forth are same for the SOAP-based requests as they are for the RESTful requests of the same type.

6. The entrance point for the service enabler marks the beginning of the application-initiated transaction.

7. The request is sent to the Plug-in Manager.

8. The Plug-in Manager invokes the Interceptor Stack to evaluate the request. The Interceptor Stack is a flexible set of chained evaluation steps that:

   - Validates the request
   - Enforces a range of policy decisions based on SLAs (and, possibly, additional rules)
   - Performs any necessary data manipulation
   - Routes the request to an appropriate protocol translation module (a network plug-in). Routing can be done on a wide variety of parameters.

> **Note:** Should a request fail because of an unavailable module, an interceptor retries the request using one of the remaining eligible modules.

9. The request is sent to the network plug-in to be translated into the protocol suitable for the underlying network node. All state information required by the underlying network node is stored within the network plug-in.
10. The request is passed to the network.
11. When the node acknowledges the request, charging data about the completed request are recorded.
12. The transaction commits.

Typical Network-Triggered Traffic Flow

The key difference between application-initiated traffic flow and network-triggered traffic flow (other than the direction) is that the application must first indicate to Oracle Communications Services Gatekeeper that it is interested in receiving traffic from the network. It does this by *registering for (*or *subscribing to) notifications*, either by sending a request to Oracle Communications Services Gatekeeper or by having the operator set up the notification using OAM methods. For example, the application could send Oracle Communications Services Gatekeeper a request to begin receiving SMSs from the network, indicating that it is only interested in SMSs that are sent to the address “12345” and that begin with the string “blue”. In SOAP-based requests it indicates the URL of the Web Service that the application has implemented to receive these notifications back. In RESTful requests it indicates the channel to which the notifications should be published.

The registration for notifications is stored in the appropriate network plug-in, which in most cases passes it on to the underlying network node itself (in certain cases the Oracle Communications Services Gatekeeper operator must do this manually.) When a matching SMS reaches the plug-in from the network, the plug-in sends it to the Plug-in Manager, which invokes the Interceptor Stack for evaluation. Then, using RMI, the final interceptor passes the notification, along with the appropriate location from the registration, to the facade layer, which sends it on, either to the application, the channel, or, in the OSB case, to the OSB.

> **Note:** Installations that include multiple facade layers (for example, both RESTful and SOA) can be set up to use the same service enabler layer. Special configuration is required in such installations to route network-triggered traffic to the appropriate facade layer. See “Managing and Configuring the Tier Routing Manager” in *System Administrator’s Guide*, another document in this set, for more information.

See Figure 3-2 and Figure 3-3 for more information on the general traffic flow, although those figures document delivery notifications.

Platform Features

Some functionality is common to all communication services. This functionality includes:

- Service Level Agreements and Policy Enforcement
- Service Level Agreements and Network Protection
- Traffic Security
- Events, Alarms, and Charging
- Statistics and Transaction Units

Service Level Agreements and Policy Enforcement

All application access to Oracle Communications Services Gatekeeper's communication services is governed by a set of service level agreements (SLAs) between the application service provider and the Oracle Communications Services Gatekeeper operator. Oracle Communications Services Gatekeeper uses a two-tiered account grouping system to categorize application services and their providers and to simplify the creation and maintenance of SLAs:

- Service Provider Group
- Application Group

For more information on the account system, see The Administration Model, a section in this document. Out of the box, Oracle Communications Services Gatekeeper provides standard SLAs for both of these types. Custom SLAs for both types can also be created.

These SLAs define whether a member of a service provider group or application group:

- Has access to a particular communication service, which can be regulated down to supported methods and parameters
- Participates in any quality of service (QoS) agreements such as:
  - Specifying the guaranteed number of requests a service provider may send through a particular communication service in a given period of time. These guarantees may be modulated by:
    - Time of Day/Day of Week

- Rate (Invocations per time period)
- Quota (Aggregated number of invocations)

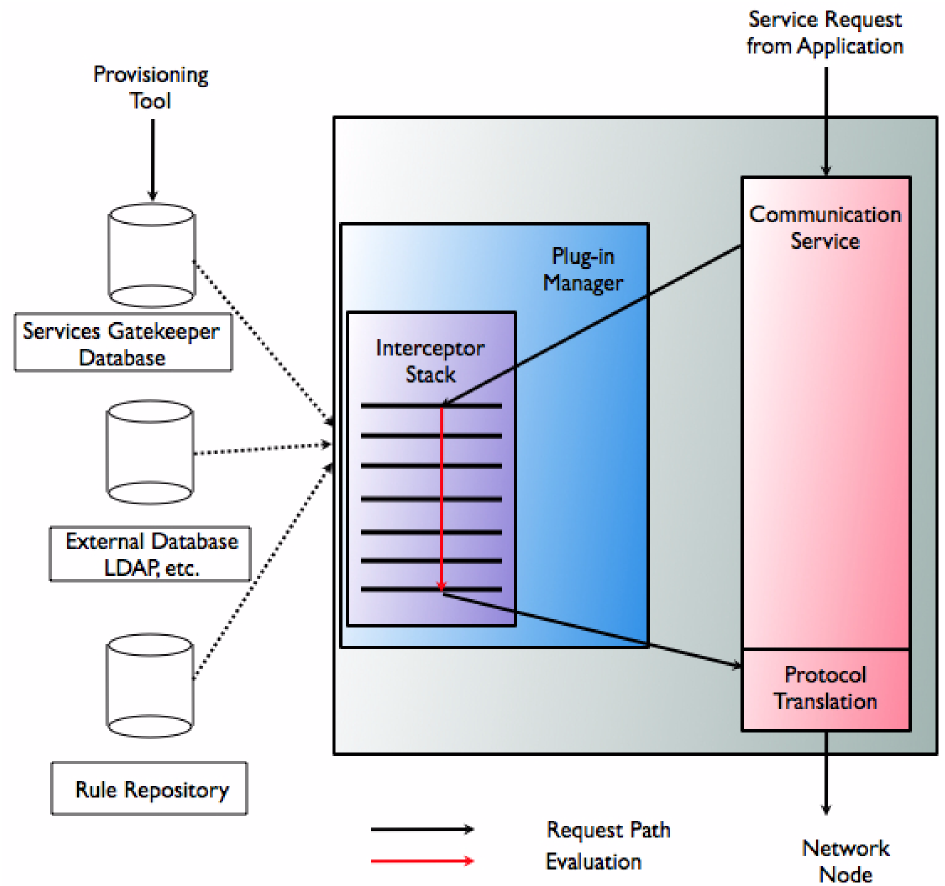
For a more detailed look at Service Provider and Application SLA structure, see "Defining Service Provider Level and Application Level Service Agreements" in *Managing Accounts and SLAs.* For a communication service-focused description, see the respective communication service chapters in *Communication Service Reference.* These books are separate documents in this set.

SLA enforcement for communication services is provided by the Interceptor Stack. As in previous versions of Oracle Communications Services Gatekeeper, it is also possible to create extended *rules* that are evaluated using the external policy engine, which is called from an interceptor.

Note: These rules represent operator specific policies defined by the operator and implemented by Oracle or a selected partner.

A simplified version of the flow is illustrated in Figure 4-2 below.

Figure 4-2 Simplified Communication Service Policy Execution Flow



Network-triggered requests are also evaluated using the Interceptor Stack.

Service Level Agreements and Network Protection

There are also SLAs that help protect the underlying network node by setting priorities for sending requests, on the level of a particular service provider group or of Oracle Communications Services Gatekeeper as a whole. Depending on the status of the underlying network, traffic can be throttled and shaped. If a particular node is overloaded, lower-priority traffic can be rejected altogether. For general information on these traffic-based (called *Node*) SLAs, see The Administration Model in this document. For more detailed information, see the "Defining Global

Node and Service Provider Group Node SLAs" chapter in *Managing Accounts and SLAs*. Out of the box, Oracle Communications Services Gatekeeper provides standard SLAs for both these types. Custom SLAs for the Global Node type can also be created.

Traffic Security

For SOAP-based Web Services interfaces, Oracle Communications Services Gatekeeper uses special SOAP headers to authenticate service provider applications. These headers are documented in the WS-Policy section of each interface's WSDL file. Processing is managed by WebLogic Server's WS-Security, which supports plaintext or digest passwords, X.509 certificates, or SAML tokens for authentication. To guarantee the confidentiality of communication between Oracle Communications Services Gatekeeper and the application, traffic can be encrypted - fully or partially - using W3C's standard XML encryption. Message integrity can be assured using the W3C XML digital signature standard. The WS-Policy section of the published WSDL for each interface describes if and how either of these standards is being used. For more information on WebLogic Server's capabilities, see *Securing WebLogic Web Services*, a document in the WLS set. Access to a particular communication service is based on the two types of SLAs discussed in Service Level Agreements and Policy Enforcement.

For RESTful Web Services interfaces, Oracle Communications Services Gatekeeper supports HTTP basic authentication, using username/password. SSL is required.

In addition, if the underlying network node provides an authentication interface, Oracle Communications Services Gatekeeper protocol plug-ins can register with it and be authenticated, making the request's transfer to the network secure.

Note: This capability is highly dependent on the protocol and the specific implementation in the node and the plug-in.

Events, Alarms, and Charging

All Oracle Communications Services Gatekeeper modules can produce general events, alarms and charging events. General events are expected system occurrences that are of importance to the operator but do not need corrective action. Alarms are system occurrences that are unexpected and may require corrective action. Charging events are the basis for CDRs, the records that provide the information needed to charge for services. CDRs are written only when the transaction that brackets the request's flow through the Network Tier commits. For more information on events and charging, see "Events, Alarms, and Charging" in *Communication Service Reference*. For more information on alarms, see *Handling Alarms*. These books are separate documents in this set.

Statistics and Transaction Units

Usage costs for Oracle Communications Services Gatekeeper are based on a maximum allowed rate (measured in *transaction units per second* or TUPS) during a specific time period per 24-hour interval. Two TUPS rates are measured: Base Platform - the more general rate - and Oracle Module - which covers only Oracle Communications Services Gatekeeper-supplied communication services. For more information on how these statistics are gathered, see *Licensing*, another document in this set.

Developing Applications

This chapter introduces developing client applications to interact with Oracle Communications Services Gatekeeper:

- Overview of Interfaces
- References
- SDK

Overview of Interfaces

Oracle Communications Services Gatekeeper allows operators to provide client application developers with a choice of interface types, based on the needs of their applications. Oracle Communications Services Gatekeeper provides:

- SOAP-based interfaces, for both traditional Web Services and Oracle Service Bus environments
- RESTful interfaces
- Native telephony interfaces (MM7 and SMPP).

SOAP-Based Interfaces

The SOAP-based Web Services APIs are based on the Parlay X 2.1 and 3.0 standards and also include three additional Extended Web Services ones to cover Binary SMS, Subscriber Profile, and WAP Push, functionality which is not supported by Parlay X. These interfaces include:

- Third Party Call (Parlay X 2.1 and 3.0)

  Using the communication services based on these interfaces, an application can set up a call between two parties (the caller and the callee), poll for the status of the call, and end the call. In addition, applications that use the Parlay X 3.0 based communication service can also add or delete additional parties, transfer call participants, get the call session information associated with a call participant, and interact with certain other communication service functionality, using Audio Call (play media to one or more parties and/or collect data from them) and Call Notification (respond to call event notifications previously established).

- Audio Call (Parlay X 3.0)

  Using the communication service based on this interface, an application can play audio to one or more call participants in a call session that was set up using the Parlay X 3.0 Third Party Call service. It is also possible to collect digits from the participant in response to the audio, which can be delivered to the application using the Parlay X 3.0 Call Notification communication service.

- Call Notification (Parlay X 2.1 and 3.0)

  Using the communication services based on these interfaces, an application can set up and end notifications on call events, such as a callee in a third-party call attempt being busy. If desired, the application can then reroute the call to another party. In addition, the Parlay X 3.0 communication service can work in concert with certain other communication service functionality, using Third Party Call or Audio Call.

- Short Messaging (Parlay X 2.1)

  Using the communication service based on this interface, an application can send SMS text messages, ringtones, or logos to one or multiple addresses, set up and receive notifications for final delivery receipts of those sent items, and arrange to receive SMSs meeting particular criteria from the network.

- Multimedia Messaging (Parlay X 2.1)

  Using the communication service based on this interface, an application can send multimedia messages to one or multiple addresses, set up and receive notifications for final delivery receipts of those sent items, and arrange to receive MMSs meeting particular criteria from the network.

- Terminal Location (Parlay X 2.1)

Using the communication service based on this interface, an application can request the position of one or more terminals or the distance between a given position and a terminal. It can also set up and receive notifications based on geographic location or time intervals.

- Presence (Parlay X 2.1)

  Using the communication service based on this interface, an application can act as either of two different parties to a presence interaction: as a *presentity* or as a *watcher.* A presentity agrees to have certain data (called attributes) such as current activity, available communication means, and contact addresses made available to others while a watcher is a consumer of such information. As a watcher, an application can request to subscribe to all or a subset of a presentity's data, poll for that data, and start and end presence notifications. As a presentity, an application can publish presence data about itself, check to see if any new watchers wish to subscribe to its presence data, authorize those watchers it chooses to authorize, block those it wishes not to have access, and get a list of currently subscribed watchers.

- Payment (Parlay X 3.0)

  Using the communication service based on this interface, an application can charge an amount to an end-user's account using Diameter, refund amounts to that account, and split charge amounts among multiple end-users. An application can also reserve amounts, reserve additional amounts, charge against the reservation or release the reservation.

- Binary SMS (EWS)

  Using the communication service based on this interface, an application can send and receive generic binary objects (for example, a vCard) using SMS mechanisms, and set up and receive notifications. This interface is not based on the Parlay X standards, but instead belongs to the Oracle Extended Web Services set.

- WAP Push (EWS)

  The application-facing interface of this communication service is not based on the Parlay X 2.1 specification. Many elements within it, however, are based on widely distributed standards. Using the communication service based on this interface, an application can send a WAP Push message, send a replacement WAP Push message, or set up status notifications about previously sent messages.

- Subscriber Profile (EWS)

  The application-facing interface of this communication service is based on a subset of that in a proposed Parlay X version. Using the communication service based on this interface, an application can retrieve either individual properties associated with a subscriber profile record stored in an LDAP data source in the underlying network or entire profiles from that data source.

- Session Manager (EWS)

  Using this communication service, an application can establish an Oracle Communications Services Gatekeeper session.

RESTful Interfaces

The RESTful APIs provide access to functionality similar to the SOAP Facade. The interfaces include:

- Short Messaging

  Using the communication service based on this interface enables an application to send an SMS, a ringtone, or a logo, and to fetch SMSs and delivery status reports for the application that have been received and stored on Oracle Communications Services Gatekeeper. It also allows an application to start and stop a notification.

- Multimedia Messaging

  Using the communication service based on this interface enables an application to send an MMS and to fetch information on MMSs for the application that have been received and stored on Oracle Communications Services Gatekeeper. It also allows the application to fetch those messages. The application can also get delivery status on sent messages, and start and stop a notification

- Terminal Location

  Using the communication service based on this interface enables an application to get a location for an individual terminal or a group of terminals; to get the distance of the terminal from a specific location; and to start and stop notifications, based on geographic location or on a periodic interval

- Third Party Call

  Using the communication service based on this interface enables an application to set up a call, get information on that call, cancel the call request before it is successfully completed, or end a call that has been successfully set up

- Call Notification

  Using the communication service based on this interface enables an application to set up and remove call notifications (in which the application is informed of a particular state - busy, unreachable, etc. - of the call) or to set up and remove call direction notifications (in which the application is queried for information on handling a call that is in a particular state)

- Presence

  Using the communication service based on this interface enables an application to act as either of two different parties to a presence interaction: as a presentity or as a watcher. A presentity agrees to have certain data (called *attributes*) such as current activity, available communication means, and contact addresses made available to others while a watcher is a consumer of such information. As a watcher, an application can request to subscribe to all or a subset of a presentity's data, poll for that data, and start and end presence notifications. As a presentity, an application can publish presence data about itself, check to see if any new watchers wish to subscribe to its presence data, authorize those watchers it chooses to authorize, block those it wishes not to have access, and get a list of currently subscribed watchers.

- Payment

  Using the communication service based on this interface enables an application to charge an amount to an end-user's account using Diameter, refund amounts to that account, and split charge amounts among multiple end-users. An application can also reserve amounts, reserve additional amounts, charge against the reservation or release the reservation.

- WAP Push

  Using the communication service based on this interface enables an application to send a WAP Push message.

- Session Manager

  Using the communication service based on this interface enables an application to get a session ID, get the time remaining in a session's lifetime, and destroy a session

Native Interfaces

- Native MM7

  The application-facing interface of this communication service is based on the 3GPP MM7 standard. Using the communication service based on this interface, an application can send and receive MMSs and receive status notifications about previously sent messages.

- Native SMPP

The application-facing interface of this communication service is based on the SMS Forum standard. Using the communication service based on this interface, an application can send and receive SMSs and receive status notifications about previously sent messages.

References

Oracle Communications Services Gatekeeper ships with *Application Development Guide* and *RESTful Application Development Guide*, separate documents in this set, which cover both the APIs themselves and some additional information an application developer needs. Because the SOAP and Restful APIs are Web Services based, applications can be developed using any environment that the developer chooses.

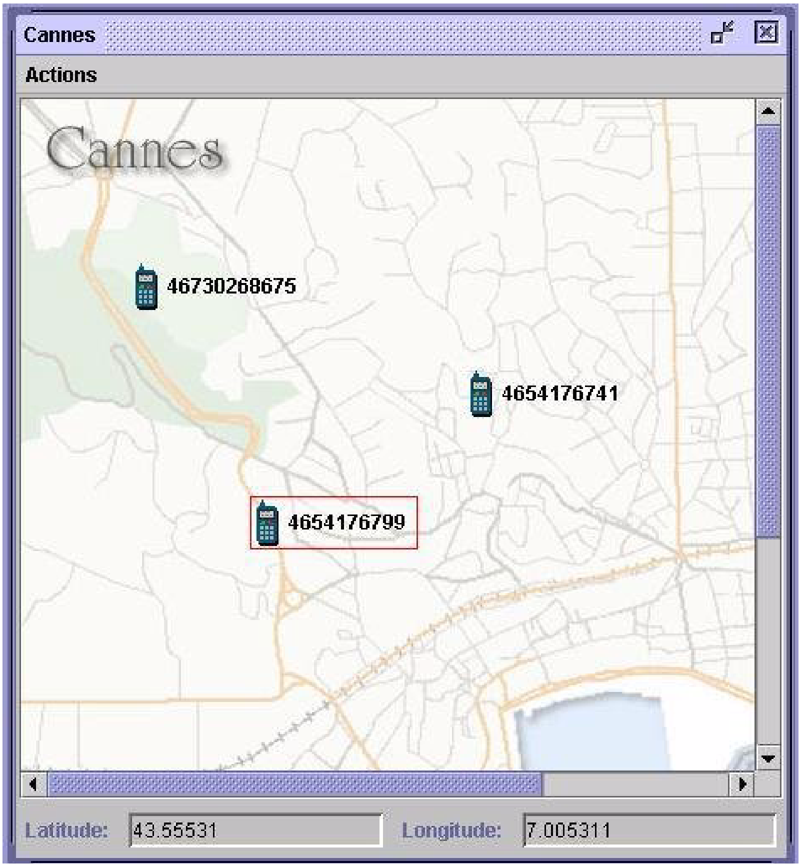
SDK

As an option, application developers using the SOAP-based interfaces can also access a set of tools created to ease the development process. The Oracle Communications Services Gatekeeper SDK includes an Oracle Communications Services Gatekeeper simulator and a network simulator. Applications use the interfaces on the Simulator just as they would on an Oracle Communications Services Gatekeeper instance running on a telecom network.

Included with the Simulator is a developer's copy of WebLogic Server, the environment in which the Simulator runs. For more on what is needed to use the SDK and run the Simulator, see the *SDK User Guide*, a separate volume in this set. The SDK also ships with a set of predefined Oracle Workshop for Weblogic controls.

In addition, there is a GUI-based testing environment that runs on the Simulator for applications that are developing MMS, SMS, WAP Push, and Terminal Location functionality. See Figure 5-1 below.

Figure 5-1 The Oracle Communications Services Gatekeeper Simulator GUI Testing Environment



The Messaging and Location testing interface of the Simulator consists of a GUI which displays a map. The map can be changed to represent different geographical areas. Mobile terminals representing the application's end users are added to the map and given a phone number. These terminals can then be used as testing targets, sending and receiving messages, and querying for location. After the terminals have been defined, they can be moved to different locations on the map.

Managing Application Service Providers

This chapter describes the framework for managing service providers and applications:

- Overview
- The Administration Model
- Partner Relationship Management Interfaces
- Other Tasks Associated with Administering Service Providers

Overview

Managing partner relationships is key to the successful convergence of third-party application services and telecom network operations. Oracle Communications Services Gatekeeper provides a partner administration model to help operators handle the needs and demands of their partners in a flexible and powerful way:

- Application service providers are registered with Oracle Communications Services Gatekeeper, by service provider account and application account.
- Each account type is associated with a group that is tied to an SLA that defines its access to both Oracle Communications Services Gatekeeper and underlying network nodes.

The service provider and application registration are performed either internally through the Oracle Communications Services Gatekeeper Management Console or through external management systems integrated using the Oracle Communications Services Gatekeeper Partner Relationship Management Interfaces or JMX.

The Administration Model

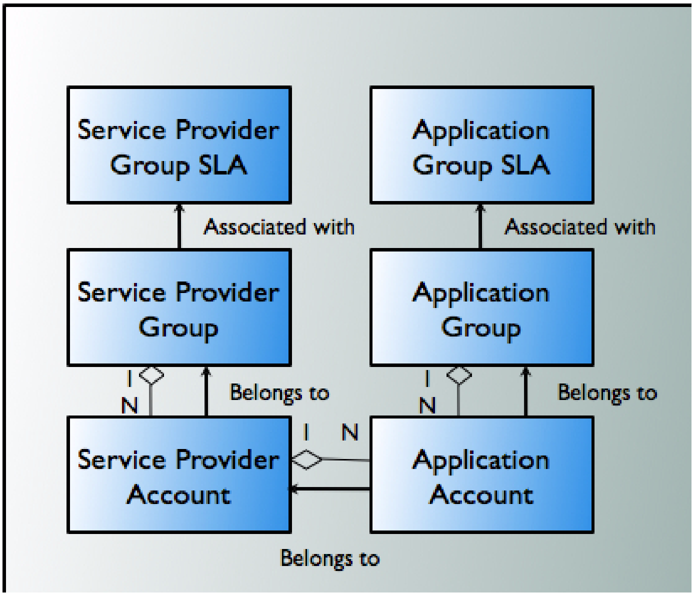
The Oracle Communications Services Gatekeeper administration model allows operators to manage application-service-provider access at increasingly granular levels of control. An application service provider registers with Oracle Communications Services Gatekeeper and is given a service provider account. To support tiering, service provider accounts belong to account groups. These account groups are then associated with their own Oracle Communications Services Gatekeeper SLAs.

Within a service provider account are individual application accounts, registered on their respective service provider accounts. As in the case of service provider accounts, these application accounts belong to account groups, each of which is associated with its own SLA.

Oracle Communications Services Gatekeeper SLAs on the service-provider and application level regulate, for example, the type of service capability made available and the maximum bandwidth use allowed. They may also specify access to charging capabilities and revenue sharing schema. Out of the box, Oracle Communications Services Gatekeeper provides support for standard versions of SLAs of both these types. Custom SLAs of both types can also be created by the operator or integrator. See Figure 6-1 for more information.

Note: Using the Platform Development Studio, integrators can extend this model to include subscribers as well. For more information, see *Platform Development Studio - Developer's Guide*, another document in this set.

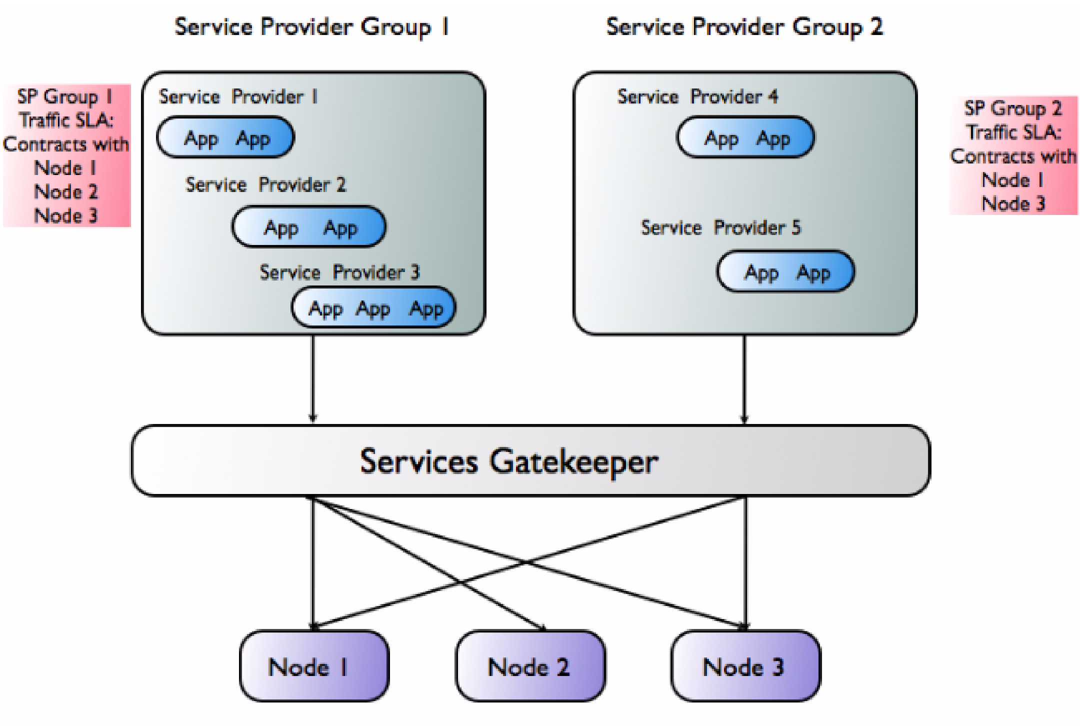
Figure 6-1 Service Provider and Application Administration Model



In addition to these account SLAs, Oracle Communications Services Gatekeeper supports two types of traffic SLAs, Service Provider Node SLAs and Global Node SLAs. Global Node custom SLAs can also be created. These are contracts designed to protect the underlying telecom network.

Service provider node SLAs regulate the relationship between a service provider group and the network nodes to which it has access. See Figure 6-2 for more information.

Figure 6-2 Service Provider Traffic SLAs



In Figure 6-2 above, service providers in service provider group 1 are allowed to access all network nodes because their service provider node SLA (valid for all service providers within the group) contains node contracts for all nodes.

Service providers in service provider group 2 are allowed to access only network nodes 1 and 3 because their service provider node SLA contains node only contracts for node 1 and 3.

The second type of traffic SLA, the Global Node SLA, regulates the overall relationship between the Oracle Communications Services Gatekeeper and the underlying nodes.

Partner Relationship Management Interfaces

The Oracle Communications Services Gatekeeper Partner Relationship Management Interfaces provide support for the automation of the traditionally work-intensive tasks related to service provider and application administration (including supporting on-boarding workflows) using request/approve. Most of the work of registration can be shifted to the service provider, allowing the operator's role to change from that of entering registration data to that of approving the registration. Large numbers of service provider and application accounts can be managed without increasing administration overhead. Service providers are also provided with a defined and structured channel to communicate desired account changes and to retrieve usage statistics for the accounts.

For a detailed description of the Partner Relationship Management Interfaces, see the document *Integration Guidelines for Partner Relationship Management for Oracle Communications Services Gatekeeper*, another document in this set.

Other Tasks Associated with Administering Service Providers

For an application to use Audio Call-based services, announcements must be recorded and installed in the network. For more information on these areas, see *System Administrator's Guide,* another document this set.

Managing Oracle Communications Services Gatekeeper: OAM

This chapter describes Operation, Administration, and Maintenance (OAM) functionality for Oracle Communications Services Gatekeeper.

- Overview
- The WebLogic and Oracle Communications Services Gatekeeper Management Console
- OAM Tasks Overview
- OSS Integration

Overview

Oracle Communications Services Gatekeeper is usually controlled through the Oracle Communications Services Gatekeeper Management Console, a specialized extension of the general WebLogic Server Administration Console. The console is a Web-based tool and can be run in any environment that supports appropriate Web browsers. For general information on the administration console, see *Introduction to Oracle WebLogic Server* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/intro/.

For some tasks, you can also use scripts that run in the WebLogic Scripting Tool: see *Oracle Weblogic Server WebLogic Scripting Tool* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/config_scripting/.

In addition, all or selected parts of the management application can be integrated with external Operation Support Systems (OSS) using JMX/JMS and alarms can be distributed using SNMP traps.

Finally, the application service provider management tool functionality can be integrated with PRM and CRM systems using the Oracle Communications Services Gatekeeper Partner Relationship Management Interfaces.

Administrative users can be divided into user groups with access to different aspects of the administrative functionality. Within user groups, individual users can have differing levels of access. See *System Administrator's Guide* for more information.

The WebLogic and Oracle Communications Services Gatekeeper Management Console

The Oracle Communications Services Gatekeeper Management Console is a Web browser-based, graphical user interface that you use to manage an Oracle WebLogic Server domain. A standard production installation for Oracle Communications Services Gatekeeper consists of at least one WebLogic Server domain.

One instance of WebLogic Server in each domain is configured as an Administration Server. The Administration Server provides a central point for managing an Oracle Communications Services Gatekeeper domain. All other server instances in the domain are called Managed Servers. In Oracle Communications Services Gatekeeper, they are divided into Access Tiers and Network Tiers. In a domain that contains only a single WebLogic Server instance, such as a development environment, that server functions as the Administration Server and both Managed Servers, that is, the Access Tier and the Network Tier. The Administration Server hosts the Administration Console, which is a Web application accessible from any supported Web browser with network access to the Administration Server. To access the console, use the following URL:

```
http://<hostname>:<port>/console
```

where `hostname` is the DNS name or IP address of the Administration Server and `port` is the port on which the Administration Server listens for requests.

OAM Tasks Overview

Use the Administration Console to:

- Configure, start, and stop Oracle Communications Services Gatekeeper instances
- Configure Oracle Communications Services Gatekeeper clusters
- Configure Oracle Communications Services Gatekeeper services, such as database connectivity (JDBC) and messaging (JMS)

- Monitor server and application performance
- View server and domain log files
- View application deployment descriptors
- Edit selected run-time application deployment descriptor elements
- Upgrade communication services
- Configure security parameters and roles

Use the Oracle Communications Services Gatekeeper-specific section (accessed through the Domain Structure tree on the left side of the Administration Console) to:

- Configure Oracle Communications Services Gatekeeper communication services
- Manage administrative users and groups
- Provision Application Service Providers, Applications, Application Instances, and related SLAs.
- Monitor alarms, CDRs, and EDRs
- Create multiple plug-in instances, set up plug-in routing, etc.

Tasks performed outside the Console

- Extend Oracle Communications Services Gatekeeper's functionality
- Back up and restore the system
- Upgrade the system

Complete information about Oracle Communications Services Gatekeeper OAM can be found in the *System Administrator's Guide*, another document in this set.

OSS Integration

- All or selected parts of the management application can also be integrated with external Operation Support Systems (OSS) through secured JMX/JMS interfaces. For more information on working with JMX, see *Oracle WebLogic Server Developing Manageable Applications with JMX* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/jmxinst/ and *Oracle WebLogic Server Configuring and Using the WebLogic Diagnostics Framework* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/wldf_configuring/. Alarm supervision systems can set up external JMS listeners to receive user definable types of event-based data, including standard alarms. SNMP traps are sent to any registered SNMP managers.

Charging and Billing Integration

The following describes Oracle Communications Services Gatekeeper's charging functionality:

- Overview
- CDR-Based Charging
- Content-Based Charging and Accounting
- Billing System Integration

Overview

Oracle Communications Services Gatekeeper makes it possible to tailor the type of charging associated with each application service. An application can use one or more of the following alternatives:

- Charging based on time used or per-use services (CDR based)
- Charging based on the content or value of the used service (CBC)

CDR-Based Charging

CDRs are used for charging based either on time used or on access to certain per-use services. Charging based on time used is typically employed for calls. Per-use might be employed, for example, to charge for a locating a terminal.

CDR data can be stored in Gatekeeper's internal charging database or retrieved in real-time by billing and post processing systems through a billing gateway (this requires integration with the billing gateway. For more information, see Billing System Integration).

Data Generation

Charging data is generated every time an application uses a communication service. The charging data is recorded by the communication service during the period the application interacts with the network. When the interaction is closed, the communication service stores the charging data as a CDR in the Oracle Communications Services Gatekeeper's database. (If Gatekeeper is integrated with a billing gateway, the charging data is sent directly to the billing gateway.)

Content-Based Charging and Accounting

Content or value based charging makes it is possible to charge an end-user based on the variable value of a used service rather than on time used or flat rates. This can be used, for example, when downloading music video clips or in m-commerce applications. Oracle Communications Services Gatekeeper supports both prepaid and postpaid end-user accounts.

Billing System Integration

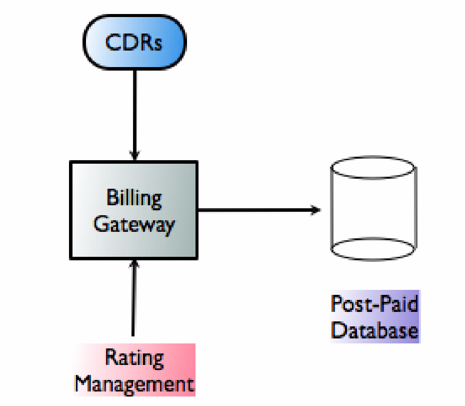
Oracle Communications Services Gatekeeper can be integrated with external billing systems, either those that receive charging data directly or those that automatically retrieve information from Oracle Communications Services Gatekeeper’s database. CDRs can be customized to fit the requirements of these systems, both in terms of format and behavior. Out of the box, Oracle Communications Services Gatekeeper has support for integration with BRM, but can be used with any system that supports Diameter, both offline (Rf), using the CDR to Diameter functionality, and online (Ro), using either the credit control interceptors for in-traffic credit checks or the Payment communication service for direct billing.

Billing Gateways

Real-time settlement of prepaid accounts using CDR-based charging requires integration through a billing gateway. This method can also be used to support postpaid services.

When integrating through a billing gateway, the billing gateway retrieves the CDRs in real-time through an external JMS-based charging listener. Rating, rating management, billing information storage, and prepaid accounts settlement are handled by the billing gateway. The flow is shown above in Figure 8-1.

Figure 8-1 Billing Integration Through Billing Gateway

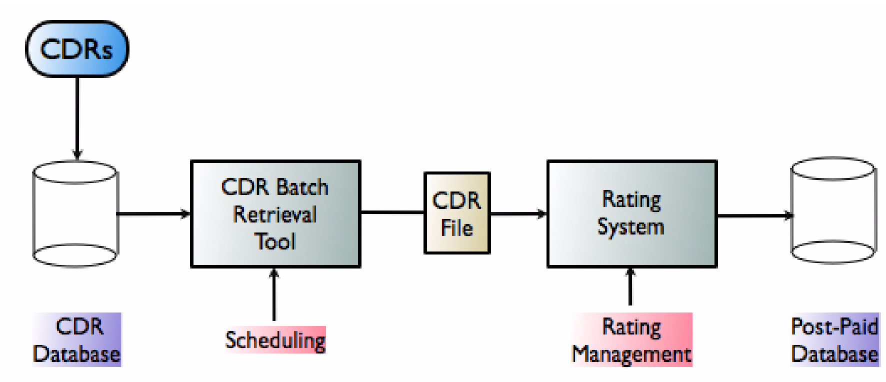


CDR Database

If an applications uses postpaid accounts, it is possible to integrate billing by retrieving CDRs that have been stored in the Oracle Communications Services Gatekeeper database.

When integrating using this method, a CDR batch retrieval tool retrieves the CDRs from the database and stores them in a file format. The CDR file is processed by a rating system that transforms it into billing information and then stores it in a post-paid accounts database. The flow is shown above in Figure 8-2.

Figure 8-2 Billing Integration Using the Database



Charging Using Diameter

Oracle Communications Services Gatekeeper has specific support for using the Diameter protocol for online (Ro) and offline (Rf) charging. The Parlay X 3.0 Payment communications services can be used for online charging using Diameter. Offline charging can be integrated using the CDR to Diameter module, which converts CDRs generated by Oracle Communications Services Gatekeeper into the appropriate format for Diameter charging requests and then sends them on to an Offline Diameter Server.

CHAPTER 9

Redundancy, Load Balancing, and High Availability

This chapter presents the redundancy, load balancing and high availability functionality in Oracle Communications Services Gatekeeper. Oracle Communications Services Gatekeeper uses both software and hardware components to support these important capabilities:

- Tiering
- Traffic Management Inside Oracle Communications Services Gatekeeper
- Registering Notifications with Network Nodes
- Network Configuration
- Geographic Redundancy

Oracle Communications Services Gatekeeper's high-availability mechanisms are supported by the clustering mechanisms made available by Oracle WebLogic Server. For general information about Oracle WebLogic Server and clustering, see *Oracle WebLogic Server Using Clusters* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/cluster/.

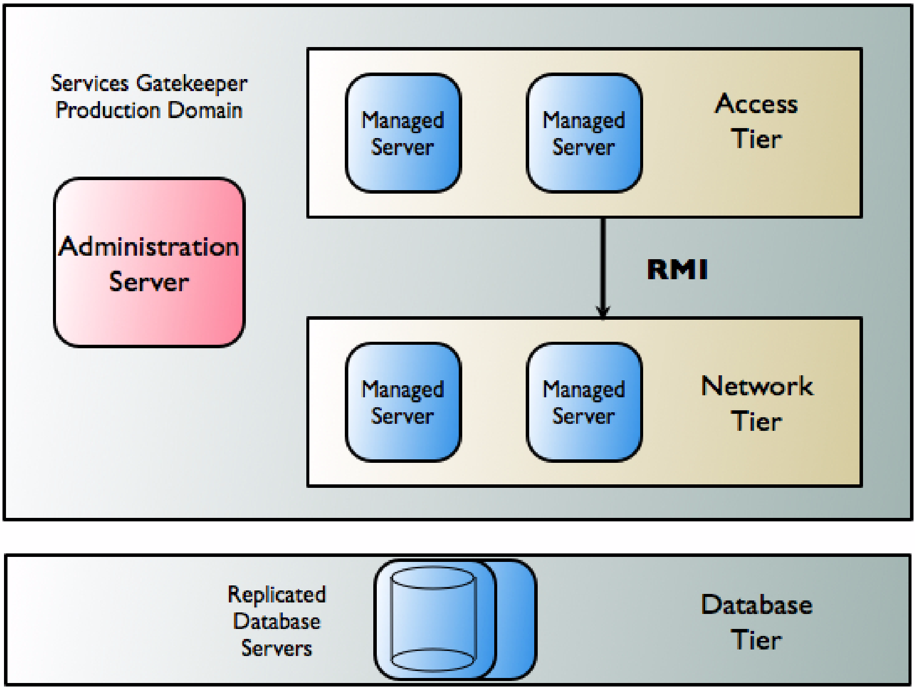
Tiering

For both high-availability and security reasons, Oracle Communications Services Gatekeeper is split into two tiers: the Access Tier and the Network Tier.

Note: Applications using the native SMPP communication service connect directly to the Network Tier.

Each tier consists of at least one cluster, with at least two server instances per cluster, and all server instances run in active mode, independently of each other. The servers in all clusters are, in the context of Oracle WebLogic Server, Managed Servers. Together the clusters make up a single WebLogic Server administrative domain, controlled through an Administration Server.

Figure 9-1 Sample Production Domain



Communication between the Access Tier and the Network Tier takes place using Java RMI. Application requests are load balanced between the Access Tier and the Network Tier and failover mechanisms are present between the two. See Traffic Management Inside Oracle Communications Services Gatekeeper for more information on these mechanisms in application-initiated and network-triggered traffic flows.

There is an additional tier containing the database. Within the cluster, data is made highly available using a cluster-aware storage service which ensures that all state data is made available across all Network Tier instances.

Traffic Management Inside Oracle Communications Services Gatekeeper

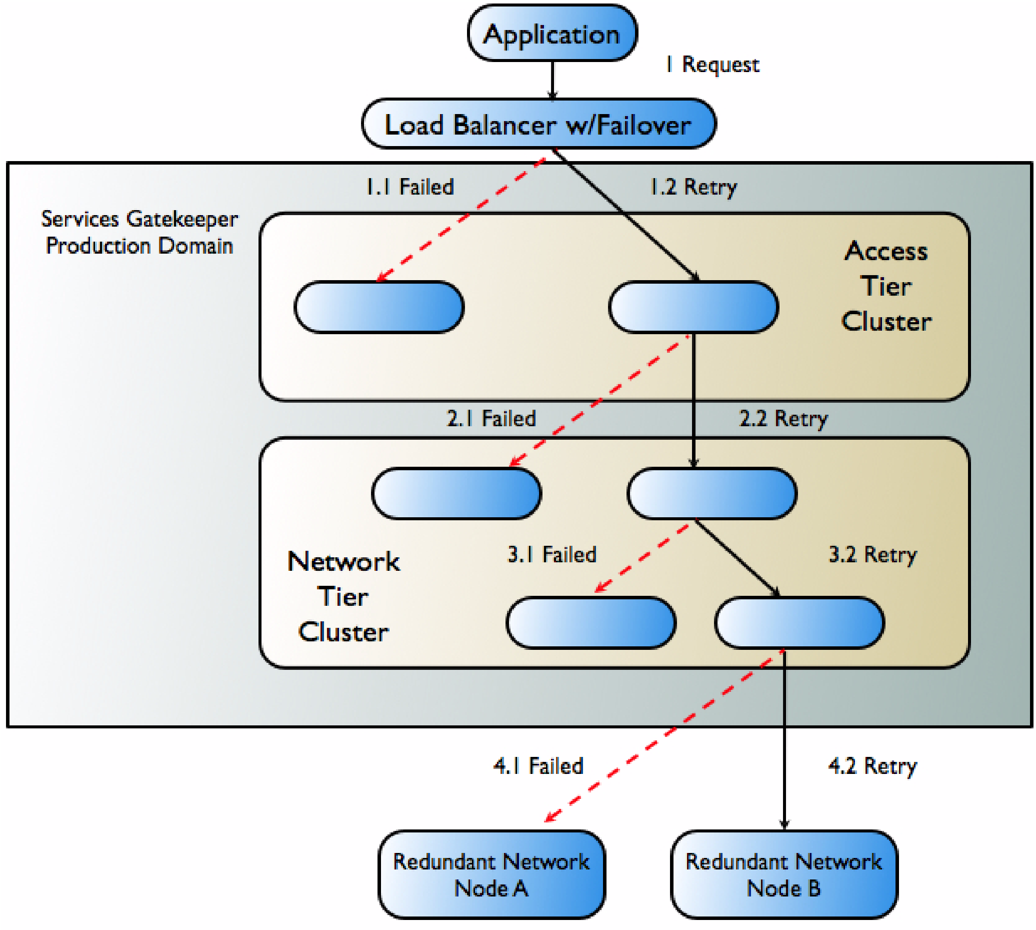
Potential failure is possible at many stages in traffic workflow in Gatekeeper. The following sections detail, tier by tier, how Oracle Communications Services Gatekeeper deals with problems that might arise in both application-initiated and network-triggered traffic.

Application-Initiated Traffic

Application-initiated traffic consists of all requests that travel from applications through Oracle Communications Services Gatekeeper to underlying network nodes.

Figure 9-2 below follows the worst-case scenario for application-initiated traffic as it passes through Oracle Communications Services Gatekeeper and the failover mechanisms that attempt to keep the request alive.

Figure 9-2 Failover Mechanisms in Application-Initiated Traffic



1. The application sends a request to Oracle Communications Services Gatekeeper. In a production environment, this request is routed through a hardware load balancer, usually protocol-aware. If the request towards the initial Access Tier server fails (1.1 in Figure 9-2), either a time out or a failure is reported. The load-balancer, or the application itself, is responsible for retrying the request.

2. The request is retried on a second server in the cluster (1.2) and it succeeds. That server then attempts to send the request on to the Network Tier.
3. The request either fails to reach the Network Tier or fails during the process of marshalling/unmarshalling the request as it travels to the Network Tier server (2.1).
4. A failover mechanism in the Access Tier sends the request to a different server in the Network Tier cluster and it succeeds (2.2). That server then attempts to send the request on to the network node.
5. The request is sent to a plug-in in the Network Tier that is unavailable (3.1). An Interceptor from the stack retries the remaining eligible plug-ins in the same server and succeeds (3.2).
6. The attempt to send the request to the telecom network node fails (4.1).
7. If a redundant pair of network nodes exists, the request is forwarded to the redundant node (4.2). If this request fails, the failure is reported to the application.

Note: In addition to the mechanisms described above, Oracle Communications Services Gatekeeper also allows the creation of multiple instances of a single SMPP plug-in type, with multiple binds, which can set up redundant connections to one or more network nodes. Such mechanisms can also increase throughput, and help optimize traffic to SMSCs with small transport windows.

Network-triggered Traffic

Network-triggered traffic can consist of the following:

- Requests that contain a payload, such as terminal location or an SMS
- Acknowledgements from the underlying network node that an application-initiated request has been processed by the network node itself. A typical example might indicate that an SMS has reached the SMSC. From an application’s perspective, this is normally processed as part of a synchronous request, although it may be asynchronous from the point of view of the network.
- Acknowledgements from the underlying network node that the request has been processed by the destination end-user terminal; for example, an SMS delivery receipt indicating that the SMS has been delivered to the end-user terminal. From an application’s perspective, this is normally handled as an incoming notification.

For network-triggered traffic, Oracle Communications Services Gatekeeper relies on internal mechanisms in concert with the capabilities of the telecom network node or other external artifacts such as load-balancers with failover capabilities to do failover.

Some network nodes can handle the registration of multiple callback interfaces. In such cases, Oracle Communications Services Gatekeeper registers one primary and one secondary callback interface. If the node is unable to send a request to the network plug-in registered as the primary callback interface, it is responsible for retrying the request by sending it to the plug-in that is registered as the secondary callback interface. This plug-in resides in another Network Tier instance. The plug-ins themselves are responsible for communicating with each other and making sure that both callback interfaces are registered. See When the Network Node Supports Primary and Secondary Notification for more information.

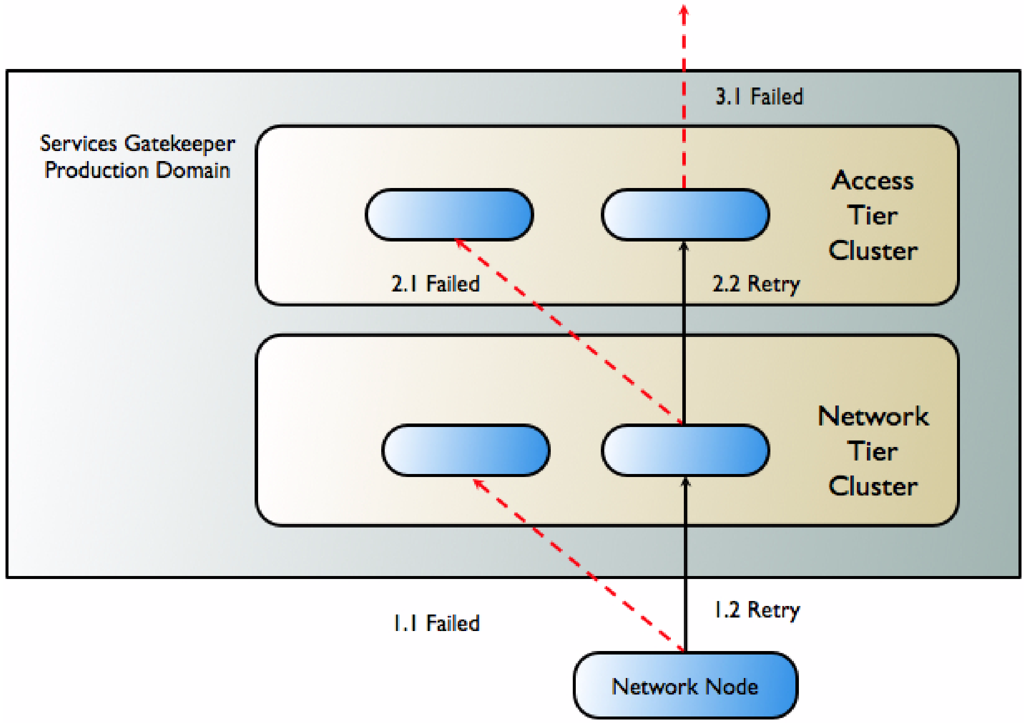
In the case of communication services using SMPP, all Oracle Communications Services Gatekeeper plug-ins can function equally as receivers for any transmission from the network node.

Finally, for HTTP-based protocols, such as MM7, MLP, and PAP, Oracle Communications Services Gatekeeper relies on an HTTP load balancer with failover functionality between the telecom network node and Oracle Communications Services Gatekeeper. See When the Network Node Supports Only Single Notification for more information.

If a telecom network protocol does not support load balancing and high availability, a single point of failure is unavoidable. In this case, all traffic associated with a specific application is routed through the same Network Tier server and each plug-in has one single connection to one telecom network node.

The worst-case scenario for network-triggered traffic for medium life span notifications using a network node that supports primary and secondary callback interfaces is described in Figure 9-3.

Figure 9-3 Failover Mechanisms in Network-Triggered Traffic



1. A telecom network node sends a request to the Oracle Communications Services Gatekeeper network plug-in that has been registered as the primary. It fails (1.1 in Figure 9-3) due to either a communication or server failure.
2. The telecom network node resends the request, this time to the plug-in that is registered as the secondary callback interface (1.2). This plug-in is in a different server instance within the Network Tier cluster. It succeeds.
3. The Network Tier attempts to send the message to the callback mechanism in the Access Tier. It fails (2.1).
4. If the request fails to reach the Access Tier, or failure occurs during the marshalling/unmarshalling process (2.1), the Network Tier retries, targeting another server in the Access Tier. It succeeds (2.2).

Note: If, however, the failure occurs after processing has begun in the Access Tier, failover does not occur and an error is reported to the network node.

5. The callback mechanism in the Access Tier attempts to send the request to the application (3.1). If the application is unreachable or does not respond, the request is considered as having failed, and an error is reported to the network node.

Registering Notifications with Network Nodes

Before applications can receive network-triggered traffic, or notifications, they must register their interest in doing so with Oracle Communications Services Gatekeeper, either by sending a request or having the operator set the notification up using OAM methods. In turn, these notifications must be registered with the underlying network node that will be supplying them. The form of this registration is dependent on the capabilities of that node.

If registration for notifications is supported by the underlying network node protocol, the communication service's network plug-in is responsible for performing it, whether the registration is the result of an application-initiated registration request or an online provisioning step in Oracle Communications Services Gatekeeper. For example, all OSA/Parlay Gateway interfaces support such registration for notifications.

Note: Some network protocols support some, but not all registration types. For example, in MM7 an application can register to receive notifications for delivery reports on messages sent *from* the application, but not to receive notifications on messages sent *to* the application from the network. In this case, registration for such notifications can be done as an off-line provisioning step in the MMSC.

Whether the plug-in sets up the notification in the network or it is done using OAM, Oracle Communications Services Gatekeeper is responsible for correlating all network-triggered traffic with its corresponding application.

Notification Life Span

Notifications are placed into three categories, based on the expected life span of the notification. These categories determine the failover strategies used:

- Short life span

  These notifications have an expected life span of a few seconds. Typically these are delivery acknowledgements for hand-off of the request to the network node, where the response to the request is reported asynchronously. For this category, a single plug-in, the originating one, is deemed sufficient to handle the response from the network node.

- Medium life span

  These notifications have an expected life span of minutes up to a few days. Typically these are delivery acknowledgements for message delivery to an end-user terminal. For this category, the delivery notification criteria that have been registered are replicated to exactly one additional instance of the network protocol plug-in. The plug-in that receives the notification is responsible for registering a secondary notification with the network node, if possible.

- Long life span

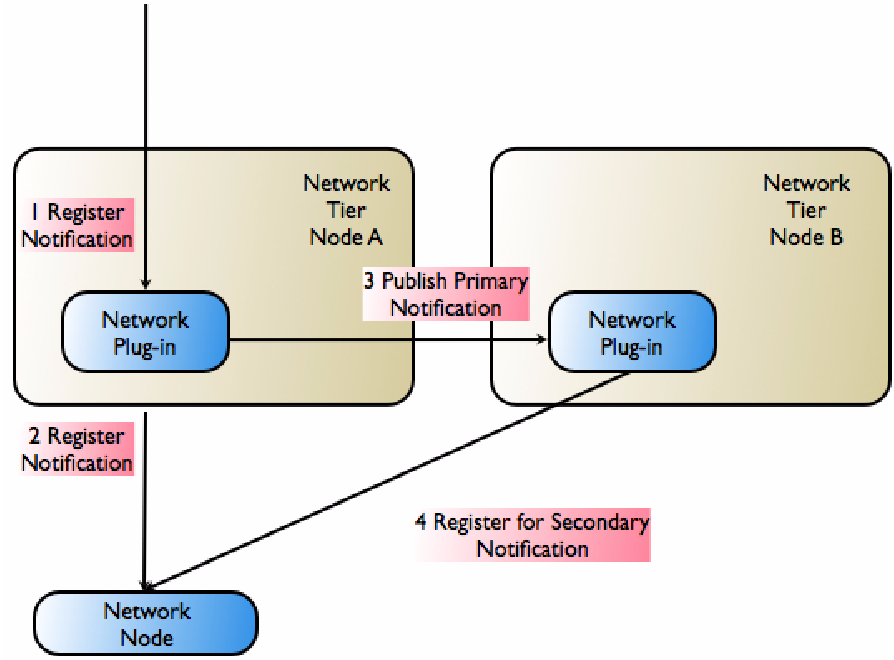
  These notifications have an expected life span of more than a a few days. Typically these are registrations for notifications for network-triggered SMS and MMS messages or calls that need to be handled by an application. For this category, the delivery notification criteria are replicated to all instances of the network plug-in. Each plug-in that receives the notification is responsible for registering an interface with the network node.

When the Network Node Supports Primary and Secondary Notification

Figure 9-4 illustrates how Oracle Communications Services Gatekeeper registers both primary and secondary notifications with network nodes that support it. This capability must be supported by the network protocol in the abstract and in the implementation of the protocol as it exists in both the network node and the communication service's network plug-in.

Note: The scenario assumes that the network node supports registration for notifications with overlapping criteria (primary/secondary).

Figure 9-4 Network Node Supports Primary/Secondary Notifications



1. The request to register for notifications enters the network protocol plug-in from the application.
2. The primary notification is registered with the telecom network node.
3. The notification information is propagated to another instance of the network protocol plug-in.
4. The secondary notification is registered with the telecom network node.

Note: The concept of primary/secondary notification is not necessarily ordered. The most recently registered notification may, for example, be designated the primary notification.

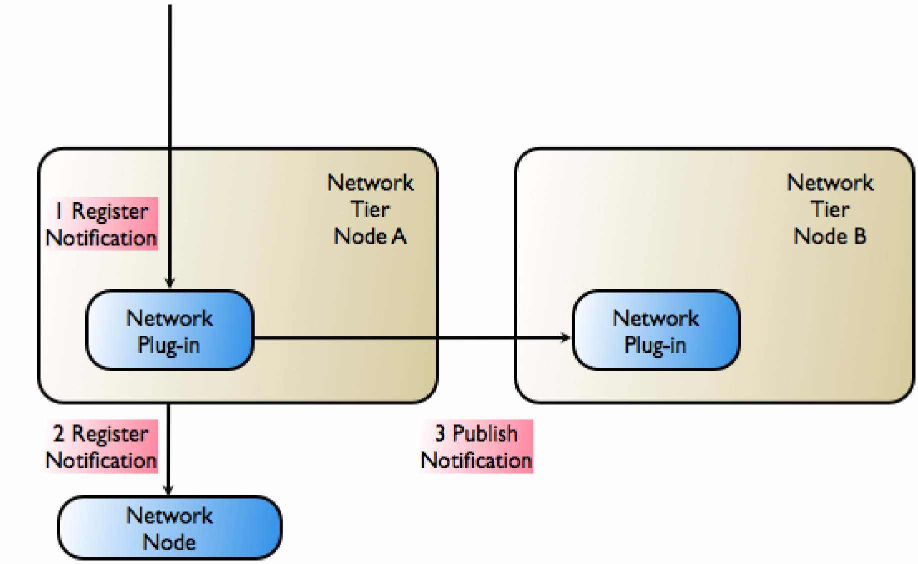
When a network-triggered request that matches the criteria in a previously registered notification reaches the telecom network node, the node first tries the network plug-in that registered the primary notification. If that request fails, the network node has the responsibility of retrying, using the plug-in that registered the secondary notification. The secondary plug-in will have all necessary information to propagate the request through Oracle Communications Services Gatekeeper and on to the correct application.

When the Network Node Supports Only Single Notification

Figure 9-5 illustrates the registration step in Oracle Communications Services Gatekeeper if the underlying network node does not support primary/secondary notification registration.

Note: The scenario assumes that the network node does not support registration for notifications with overlapping criteria. Only one notification for a given criteria is allowed.

Figure 9-5 Network Node Supports Only Single Notification



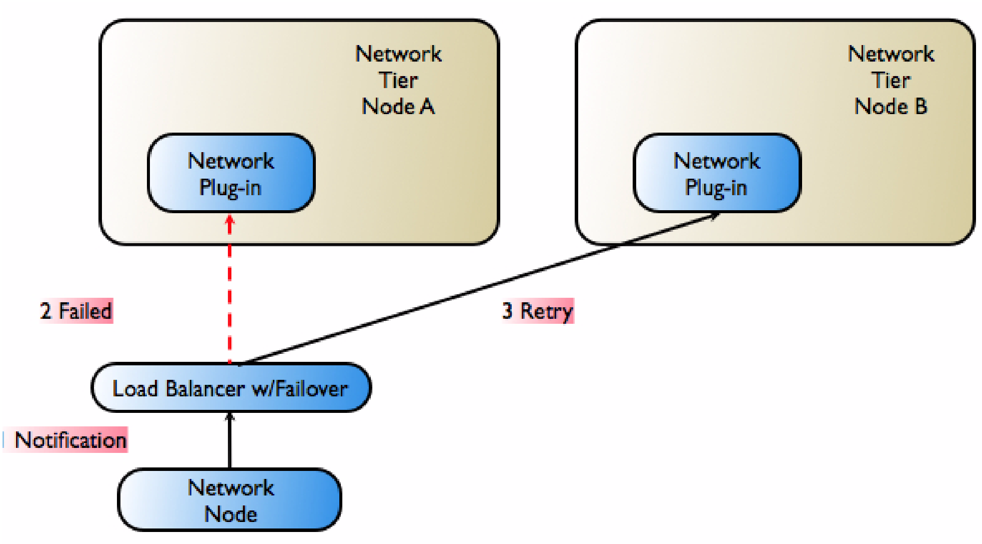
1. The request to register for notifications enters the network protocol plug-in from the application.
2. The notification is registered with the telecom network node.
3. The notification information (matching criteria, target URL, etc.) is propagated to another instance of the network protocol plug-in. The plug-in makes the necessary arrangements to be able to receive notifications.

There are two possibilities for high-availability and failover support in this case:

- All plug-ins can receive notifications from the network node. This is the case with SMPP, in which all plug-ins can function as receivers for any transmission from the network node.
- A load balancer with failover support is introduced between the network protocol plug-in and the network node. This is the case with HTT- based protocols, as in Figure 9-6.

  Note: Whether or not this is possible depends on the network protocol, because the load-balancer must be protocol aware.

Figure 9-6 Traffic With a Single Notification Only Node: Load-Balancer With Failover Support



Network Configuration

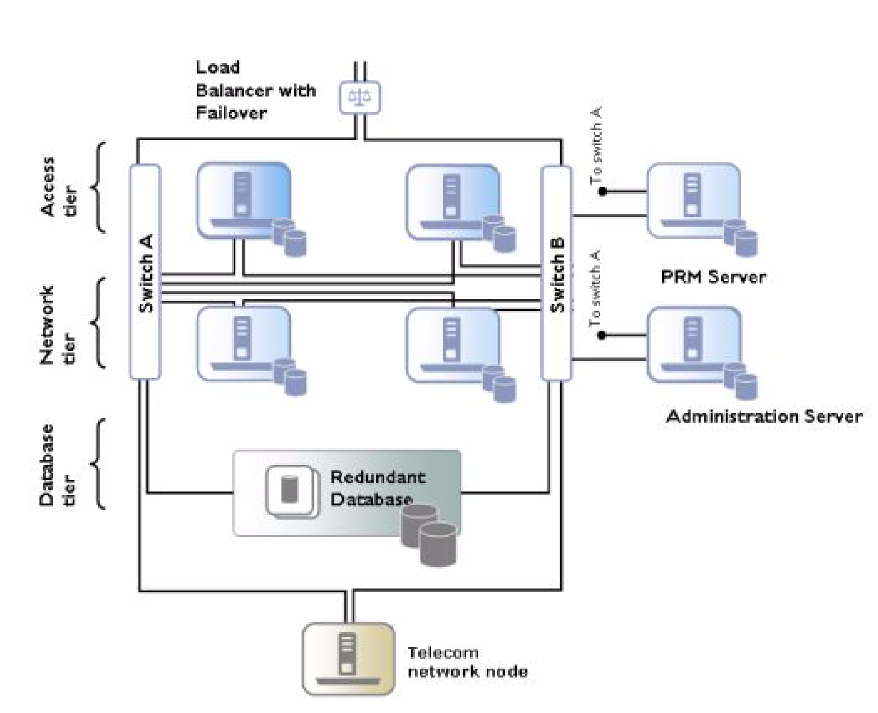
The general structure of a production Oracle Communications Services Gatekeeper installation is also designed to support redundancy and high availability. A typical installation consists of a number of UNIX/Linux servers connected through duplicated switches. Each server has redundant network cards connected to separate switches. The servers are organized into clusters, with the number of servers in the cluster determined by the needed capacity.

As described previously, Oracle Communications Services Gatekeeper is deployed on an Access Tier, which manages connections to applications, and a Network Tier, which manages connections to the underlying telecom network. For security, the Network Tier is usually connected only to Access Tier servers, the appropriate underlying network nodes, and the Oracle WebLogic Server Administration Server, which manages the domain. A third tier hosts the database. This tier should be hosted on dedicated, redundant servers. For physical storage, a Network Attached Storage using fibre channel controller cards is an option.

Because the different tiers perform different tasks, their servers should be optimized with different physical profiles, including amount of RAM, disk-types, and CPUs. Each tier scales individually, so the number of servers in a specific tier can be increased without affecting the other tiers.

A sample configuration is shown in Figure 9-7. Smaller systems in which the Access Tier and the Network Tier are co-located in the same physical servers are possible but only for non-production systems. Particular hardware configurations depend on the specific deployment requirements and are worked out in the dimensioning and capacity planning stage.

Figure 9-7 Sample Hardware Configuration



In high-availability mode, all hardware components are duplicated, eliminating single point of failure. This means that there are at least two servers executing the same software modules, that each server has two network cards, and that each server has a fault-tolerant disk system, as, for example, RAID.

The administration server may have duplicate network cards, connected to each switch.

For security reasons, the servers used for the Access Tier can be separated from the Network Tier servers using firewalls. The Access Tier servers reside in a Demilitarized Zone (DMZ) while the Network Tier servers are in a trusted environment.

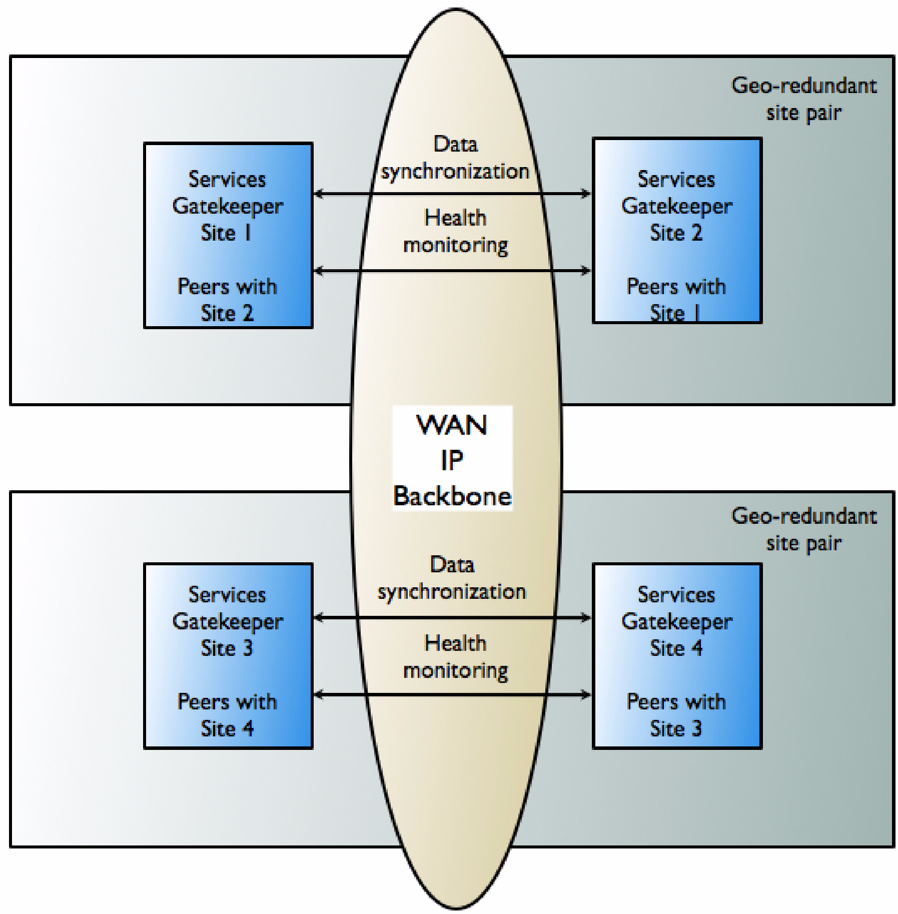
Geographic Redundancy

All Oracle Communications Services Gatekeeper modules in production systems are deployed in clusters to ensure high availability. This prevents single points of failure in general usage. Within a cluster, a Budget Service cluster-local master regulates the enforcement of SLAs. The enforcement service is highly available and is migrated to another server should the cluster-local master node fail. See “Managing and Configuring Budgets” in *System Administrator’s Guide* for more information on this mechanism.

However, to prevent service failure in the face of catastrophic events - natural disasters or massive system outages like power failures - Oracle Communications Services Gatekeeper can also be deployed at two geographically distant sites that are designated as site pairs. Each site, which is a Oracle Communications Services Gatekeeper domain, has another site as its peer. See Figure 9-8 an overview. Application and service provider configuration information, including related SLAs and budget information, is replicated and enforced across sites.

Note: Custom, Subscriber, Service Provider Node, and Global Node SLAs cannot be replicated across sites.

Figure 9-8 Overview of Geographically Redundant Site Pairs



Geo-Redundant Sites

In a geo-redundant setup, all sites have a geographic site name and each site is configured to have a reference to its peer site using that name. The designated set of information is synchronized between these site peers.

One site is defined as the geomaster, the other as the slave. Checks are run periodically between the site pairs to verify data consistency and an alarm is triggered if mismatches are found, at which point the administrator can force the slave to re-sync to the geomaster, using the `syncFromGeoMaster` operation. Any relevant configuration changes made to either site are written synchronously across the site pairs, so that a failure to write to either the geomaster or the slave causes the write to fail and an alarm to fire.

During the period in which the slave is syncing up with the geomaster, both the geomaster and the slave sites are in read-only mode. No configuration changes can be made. If a slave site becomes unavailable for any reason, the geomaster site becomes read-only either until the slave site is available and has completed all data replication, or until the slave site has been removed from the geomaster site's configuration, terminating geo-redundancy.

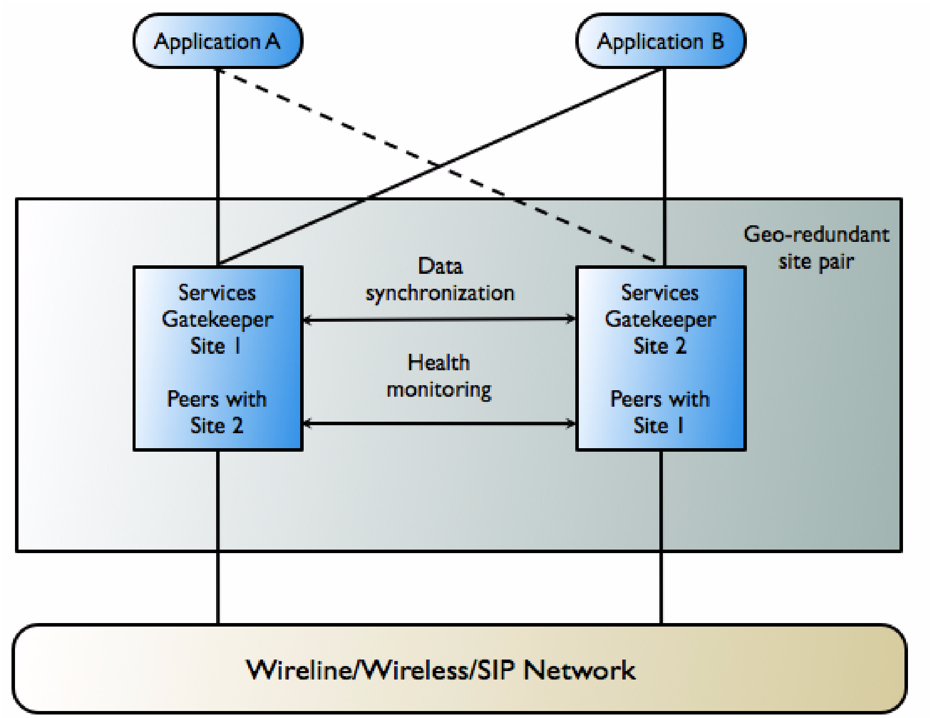
Note: If a new site is then added to replace the terminated site, it must be added as a slave site. The site that is designated the geomaster site must remain the geomaster site for the lifetime of the site configuration.

If a geomaster site fails permanently, the failed site should be removed from the configuration using the GeoRedundantService. If a replacement site is added to the configuration, the remaining operating site must be reconfigured to be the geomaster and the replacement site must be added as the slave.

Applications and Geo-Redundancy

For applications, geo-redundancy means that their traffic can continue to flow in the face of a catastrophic failure at an operator site. Even applications that normally use only a single site for their traffic can fail over to a peer site while maintaining ongoing SLA enforcement for their accounts. This scenario is particularly relevant for SLA aspects that have longer term impact, such as quotas.

Figure 9-9 Geographically Redundant Site Pairs and Applications



In many respects, the geo-redundancy mechanism is *not* transparent to applications. There is no single sign-on mechanism across sites, and an application must establish a session with each site it intends to use. In case of site failure, an application must manually fail over to a different site.

While application and service provider budget and configuration information are maintained across sites, state for *ongoing conversations* is not maintained. Conversations in this sense are defined in terms of the correlation identifiers that are returned to the applications by Oracle Communications Services Gatekeeper or passed into Oracle Communications Services Gatekeeper from the applications. Any state associated with a correlation identifier exists on only a single geographic site and is lost in the event of a site-wide disaster. Conversational state includes, but is not limited to, call state and registration for network-triggered notifications. This type of state is considered volatile, or transient, and is not replicated at the site level.

This means that conversations must be conducted and complete on their site of origin. If an application wishes to maintain conversational state cross-site - for example, to maintain a registration for network-triggered traffic - it must register with each site individually.

Note: On the other hand, this type of affinity does allow load balancing between sites for different or new conversations. For example, because each request to send an SMS message constitutes a new conversation, sending SMS messages can be balanced between the sites.

Below is a high-level outline of the redundancy functionality:

- The contractual usage relationships represented by SLAs can be enforced across geographic site domains. The mechanism covers SLAs on both the service provider group and application group level.
- Service provider and application account configuration data, including any changes to this information, can be replicated across sites, reducing the administrative overhead in setting up geo-redundant site pairs.
- When peer sites fail to establish connection a configurable number of times, a connection-lost alarms is raised.
- Alarms are also generated:
  - If there is a site configuration mismatch between the two sites; for example if site A treats site B as a peer, but site B does not recognize site A as a peer
  - If the paired sites do not have identical application and service provider configuration information, including related SLAs and budget information
  - If the master site fails to complete a configuration update to the slave site

Service Extensibility

This chapter describes how to extend the Oracle Communications Services Gatekeeper functionality, including:

- Overview
- The Platform Development Studio

Overview

Networks change. Existing functionality is parsed in new ways to support new features. New nodes with new or modified abilities are added. Because of Service Gatekeeper's highly modular design, exposing these new features to partners is straightforward. There are several ways to extend Oracle Communications Services Gatekeeper:

- Entirely new communication services
- New network plug-ins that can work with existing application facing interfaces
- New application-facing-interface types (facades) for existing network plug-ins
- New or reordered interceptors or policy rules
- Integration with the existing network mechanisms with EDR listeners and SMNP MIBs.

The Platform Development Studio

To help operators and systems integrators, Oracle Communications Services Gatekeeper ships with the Oracle Communications Services Gatekeeper Platform Development Studio (PDS). The PDS comprises the following features:

- *Platform Development Studio - Developer's Guide*

  A detailed guide covering how to:

  – Install the PDS
  – Use the Eclipse wizard to generate a project
  – Understand the example communication service
  – Use Service Gatekeeper's container services
  – Add Policy Rules
  – Create Subscriber SLAs for subscriber-based policy
  – Update EDR filters and add JMS-based listeners
  – Reorder the Interceptor Stack or create new interceptors

- *Platform Test Environment Guide*

  A detailed guide covering how to:

  – Load the Platform Test Environment (PTE)
    - In standalone mode, with a Java Swing-based GUI
    - In console mode, particularly for use with the Unit Test Framework
  – Run the Application Service Clients
  – Use the Network Simulators
  – Utilize the utilities such as the MBean browser and the JMS-based EDR listeners
  – Understand the example module and extend the PTE

- Example Communication Service, including:

  – Source code
  – WSDLs
  – Build files

  - Example unit test
  - Example Subscriber SLA/Profile Provider
- The Eclipse wizard

  The developer supplies information to an Eclipse plug-in wizard, which automatically sets up an Extension Project. Included within this project can be a substantial amount of generated code, including:

  - The entire Access Tier, with the Web Service implementation and any callback modules (EJBs) that are necessary

  Note: The Eclipse wizard supports building communication services based only on Web Services application-facing interfaces. Both SOAP and RESTful facade types are supported.

  - Most of the code for the common services code in the Network Tier
  - A skeleton of the code required for the network plug-in layer of the Network Tier
- A complete Javadoc reference
- Specialized templates and Ant tasks
- The Unit Test Framework, providing:
    - A base test class, derived from JUnit
    - Simple-to-use mechanisms for connecting to the Platform Test Environment
    - An example unit test bundled with the Communication Service Example
- The Profile Provider SPI, which allows operators and integrators to create subscriber-centric policy, associating subscribers with service provider and application groups based on individualized subscriber preferences and permissions.

Standards and Specifications

This appendix describes the specific standards that Gatekeeper supports, and provides, where possible, links to the actual specifications. A detailed statement of compliance is also available.

Application-Facing Interfaces

Parlay X 2.1

The Oracle Communications Services Gatekeeper application-facing interfaces support the following parts of the Parlay X 2.1 specification.

Note: See http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX21.html for links to the specifications.

- **Common**, ETSI ES 202 391-1 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 2).
- **Third Party Call**, ETSI ES 202 391-2 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 2: Third Party Call (Parlay X 2).
- **Call Notification**, ETSI ES 202 391-3 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 3: Call Notification (Parlay X 2).
- **Short Messaging**, ETSI ES 202 391-4 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 4: Short Messaging (Parlay X 2).
- **Multimedia Messaging**, ETSI ES 202 391-5 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 5: Multimedia Messaging (Parlay X 2).

- **Terminal Location**, ETSI ES 202 391-9 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 9: Terminal Location (Parlay X 2).
- **Presence**, ETSI ES 202 391-14 V1.2.1 (2006-12), Open Service Access (OSA); Parlay X Web Services; Part 14: Presence (Parlay X 2).

Parlay X 3.0

The Oracle Communications Services Gatekeeper application-facing interfaces support the following parts of the Draft Parlay X 3.0 specification.

Note: See http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX30.html for links to the specifications

- **Common**, Draft ETSI ES 202 504-1 v.0.0.3 (2007-06), Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 3).
- **Third Party Call**, Draft ETSI ES 202 504-2 v0.0.5 (2007-06), Open Service Access (OSA); Parlay X Web Services; Part 2; Third Party Call (Parlay X 3).
- **Call Notification**, Draft ETSI ES 202 504-3 v0.0.3 (2007-06), Open Service Access (OSA); Parlay X Web Services; Part 3: Call Notification (Parlay X 3).
- **Audio Call**, Draft ETSI ES 202 504-11 v.0.0.3 (2007-06), Open Service Access (OSA); Parlay X Web Services; Part 11: Audio Call (Parlay X 3).
- **Payment**, 3GPP TS 29.199-6 V7.2.2 (2007-06), Open Service Access (OSA); Parlay X Web Services; Part 6: Payment (Release 7)

Extended Web Services

The Extended Web Services are Oracle Communications Services Gatekeeper's proprietary application-facing interfaces. These interfaces are implementations of commonly requested functionality, including, in this release, WAP Push, Binary SMS, and Subscriber Profile. Although the interfaces themselves are not standardized, they use standardized elements.

Binary SMS

Note: See http://www.3gpp.org/ftp/Specs/html-info/23040.htm for links to the specification.

The payload, protocol identifier, and validity period rely on:

- 3GPP TS 23.040 version 6.5.0, Technical realization of Short Message Service (SMS)

WAP Push

Note: See http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html for links to the specifications.

The payload of a WAP Push message shall adhere to:

- **WAP Service Indication Specification**, as specified in Service Indication Version 31-Jul-2001 Wireless Application Protocol WAP-167-ServiceInd-20010731-a.
- **WAP Service Loading Specification**, as specified in Service Loading Version 31-Jul-2001 Wireless Application Protocol WAP-168-ServiceLoad-20010731-a.
- **WAP Cache Operation Specification**, as specified in Cache Operation Version 31-Jul-2001 Wireless Application Protocol WAP-175-CacheOp-20010731-a.

Note: The Extended Web Services WAP Push communication service does not verify the payload; it passes it on to the underlying network node.

Subscriber Profile LDAP

There is no current specifications covering subscriber-profile LDAP access, although a draft version exists. Gatekeeper's implementation is based on that draft.

RESTful APIs

There are no current specifications covering RESTful access to underlying telephony network functionality.

Native

Gatekeeper also supports some native telephony messaging application-facing interfaces. The following specifications are supported:

- **MM7**: 3GPP TS 23.140 V5.3.0 (REL-5-MM7-1-2.xsd)
- SMPP: SMPP v3.4

Network Protocol Plug-Ins

Off-the shelf, Oracle Communications Services Gatekeeper supports the network protocols listed in Table 10-1 through the use of network plug-ins. Although each plug-in is a part of a given communication service, certain protocols can be used by multiple communication services for different purposes. In these cases there may be multiple implementations of the same protocol for use in different communication services.

Table 10-1 is a list of supported network protocols organized per communication service.

Table 10-1 Network Plug-Ins Organized per Communication Service

Communication service	Network protocol plug-in	Specification
Parlay X 2.1 Third Party Call	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
	INAP/SS7	ETSI 94 INAP CS1, ETS 300 374-1, Intelligent Network (IN); Intelligent Network Capability Set 1 (CS1); Core Intelligent Network Application Protocol (INAP) http://pda.etsi.org/pda/queryform.asp
Parlay X 3.0 Third Party Call	Parlay 3.3 MultiParty Call Control	ETSI ES 201 915-4 V1.4.1 (2003-07), Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control SCF (Parlay 3), part MultiParty Call Control Service. Section MultiParty Call Control Service See http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX30.html for links to the specifications
RESTful Third Party Call	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
Parlay X 2.1 Call Notification	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
Parlay X 3.0 Call Notification	Parlay 3.3 MultiParty Call Control	ETSI ES 201 915-4 V1.4.1 (2003-07), Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control SCF (Parlay 3), part MultiParty Call Control Service. Section MultiParty Call Control Service. See http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX30.html for links to the specifications

Table 10-1 Network Plug-Ins Organized per Communication Service

Communication service	Network protocol plug-in	Specification
RESTful Call Notification	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
Parlay X 2.1 Short Messaging	SMPP v3.4	Short Message Peer to Peer, Protocol Specification v3.4, Document Version:- 12-Oct-1999 Issue 1.2. http://smsforum.net/
RESTful Short Messaging	SMPP v3.4	Short Message Peer to Peer, Protocol Specification v3.4, Document Version:- 12-Oct-1999 Issue 1.2. http://smsforum.net/
Parlay X 2.1 Multimedia Messaging	MM7 v 5.5.0	3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2 (Release 5), 3GPP TS 23.140 V5.3.0. Messages are compliant with the schema defined by one of REL-5-MM7-1-0.xsd, REL-5-MM7-1-2.xsd, or REL-5-MM7-1-5.xsd, depending on management settings. http://www.3gpp.org/ftp/Specs/html-info/23140.htm
RESTful Multimedia Messaging	MM7 v 5.5.0	3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2 (Release 5), 3GPP TS 23.140 V5.3.0. Messages are compliant with the schema defined by one of REL-5-MM7-1-0.xsd, REL-5-MM7-1-2.xsd, or REL-5-MM7-1-5.xsd, depending on management settings. http://www.3gpp.org/ftp/Specs/html-info/23140.htm

Table 10-1 Network Plug-Ins Organized per Communication Service

Communication service	Network protocol plug-in	Specification
Parlay X 2.1 Terminal Location	MLP 3.0 MLP 3.2 Only one of the above listed protocols can be used at the same moment for a given node in a domain.	Location Inter-operability Forum (LIF) Mobile Location Protocol, LIF TS 101 Specification Version 3.0.0 and Mobile Location Protocol 3.2 Candidate Version 3.2 Open Mobile Alliance, OMA-TS-MLP-V3_2-20051124-C. MLP 3.0: http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html MLP 3.2: http://www.openmobilealliance.org
RESTful Terminal Location	MLP 3.0 MLP 3.2 Only one of the above listed protocols can be used at the same moment for a given node in a domain.	Location Inter-operability Forum (LIF) Mobile Location Protocol, LIF TS 101 Specification Version 3.0.0 and Mobile Location Protocol 3.2 Candidate Version 3.2 Open Mobile Alliance, OMA-TS-MLP-V3_2-20051124-C. MLP 3.0: http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html MLP 3.2: http://www.openmobilealliance.org
Parlay X 3.0 Audio Call	Parlay 3.3 Call User Interaction and Parlay 3.3 MultiParty Call Control	ETSI ES 201 915-5 V1.4.1 (2003-07), Open Service Access (OSA); Application Programming Interface (API); Part 5: User Interaction SCF (Parlay 3). Call user interaction parts. ETSI ES 201 915-4 V1.4.1 (2003-07), Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control SCF (Parlay 3). Section MultiParty Call Control Service See http://portal.etsi.org/docbox/TISPAN/Open/OSA/ParlayX30.html for links to the specifications

Table 10-1 Network Plug-Ins Organized per Communication Service

Communication service	Network protocol plug-in	Specification
Parlay X 2.1 Presence	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
RESTful Presence	SIP	RFC 3261. http://www.ietf.org/rfc/rfc3261.txt
Parlay X 3.0 Payment	Diameter	RFC3588 http://tools.ietf.org/html/rfc3588 RFC4006 http://tools.ietf.org/html/rfc4006 3GPP TS 32.299 version 6.6.0 Release 6. http://www.3gpp.org/ftp/Specs/html-info/32299.htm
RESTful Payment	Diameter	RFC3588 http://tools.ietf.org/html/rfc3588 RFC4006 http://tools.ietf.org/html/rfc4006 3GPP TS 32.299 version 6.6.0 Release 6. http://www.3gpp.org/ftp/Specs/html-info/32299.htm
Extended Web Services WAP Push	PAP 2.0	Push Access Protocol, WAP Forum, WAP-247-PAP-20010429-a. http://www.openmobilealliance.org
RESTful WAP Push	PAP 2.0	Push Access Protocol, WAP Forum, WAP-247-PAP-20010429-a. http://www.openmobilealliance.org
Extended Web Services Binary SMS	SMPP v3.4	Short Message Peer to Peer, Protocol Specification v3.4, Document Version:- 12-Oct-1999 Issue 1.2. http://smsforum.net/

Table 10-1 Network Plug-Ins Organized per Communication Service

Communication service	Network protocol plug-in	Specification
Extended Web Services Subscriber Profile	LDAPv3	Lightweight Directory Access Protocol, RFC 4510: June 2006 http://tools.ietf.org/html/rfc4510
Native MM7	MM7 v 5.3.0	3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional description; Stage 2 (Release 5), 3GPP TS 23.140 V5.3.0. Messages are compliant with one of two schemas: either REL-5-MM7-1-2.xsd or a slightly modified version of REL-5-MM7-1-0.xsd, both available at: http://www.3gpp.org/ftp/Specs/html-info/23140.htm
Native SMPP	SMPP v3.4	Short Message Peer to Peer, Protocol Specification v3.4, Document Version:- 12-Oct-1999 Issue 1.2. http://smsforum.net/
Not applicable	Parlay 3.3 Framework This is a network-facing protocol that does not belong to a certain communication service. It is used by all Parlay plug-ins.	ETSI ES 201 915-3 V1.4.1 (2003-07), Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework (Parlay 3). http://portal.etsi.org/docbox/TISPAN/Open/OSA/Parlay33.html

Security

Oracle Communications Services Gatekeeper supports the security standards listed below. The security standards are applicable for the application-facing interfaces. Oracle Communications

Services Gatekeeper leverages Web Services Security mechanisms provided by WebLogic Server. For more information, see:

- *Securing WebLogic Server* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/secmanage
- *Securing WebLogic Web Services* at http://download.oracle.com/docs/cd/E12840_01/wls/docs103/webserv_sec/

Note: See http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss for links to the specifications.

- WS-Security Core Specification 1.1
- WS-Security 1.0 and 1.1
- UsernameToken Profile 1.1
- X.509 Certificate Token Profile 1.1
- SAML Token Profile 1.1
- SOAP Message Security 1.0
- SOAP with Attachments (SWA) 1.1

In addition, the following standards are also supported:

- WS-Addressing 1.0, http://www.w3.org/2002/ws/addr/
- WS-Policy 1.1
- WS-SecurityPolicy 1.2, http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702
- WS-Trust 1.3, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html
- WS-SecureConversation 1.3, http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html
- WS-ReliableMessaging 1.0
- WS-PolicyAttachment 1.0

Transport-level security mechanisms such as 1- or 2-way SSL or VPN tunneling can be used for the PRM interfaces.

Identity and Trust

Oracle Communications Services Gatekeeper leverages the robust identity management capabilities of Web Logic Server, including:

- Private Keys
- X.509 v3 Digital Certificates
- Symmetric & Asymmetric Key Algorithms
  - DES-CBC
  - Two-Key Triple DES
  - RC4
  - RSA
- Message Digest:
  - MD5
  - SHA
- JEE 5 & Weblogic Security Packages
  - Java Secure Socket Extension (JSSE)
  - Java Authentication & Authorization Services (JAAS)
  - Java Security Manager
  - Java Cryptography Architecture and Java Cryptography Extensions (JCE)
  - Java Authorization Contract for Containers (JACC)
  - Common Secure Interoperability Version 2 (CSIv2)