Oracle® Communications Services Gatekeeper

Deployment Guide

Release 5.1

E37527-01

June 2013

This document describes the different deployment types supported by Oracle Communications Services Gatekeeper (Services Gatekeeper).

If you are unfamiliar with some of the terms used in this guide, see *Oracle Communications Services Gatekeeper Concepts Guide*.

About Services Gatekeeper Software Components

Services Gatekeeper is built on top of Oracle WebLogic Server and can use all WebLogic Server components. It also embeds Oracle Communications Converged Application Server for connectivity to SIP networks and access to network nodes using the Diameter protocol.

Services Gatekeeper provides communication services that telecom operator in-house applications and third-party applications use to access assets in the telecom network. For a list of the supported communication services, see *Oracle Communications Services Gatekeeper Communication Service Guide*. In addition, Services Gatekeeper provides extension points and tooling that you can use to create new communication services.

Each communication service has two components:

- A service facade that exposes interfaces to be used by applications.
- A service enabler that consists of a network protocol plug-ins. The plug-ins can be instantiated. Each instance connects to a node in the telecom network using a specific protocol. These instances interact with container services as necessary.

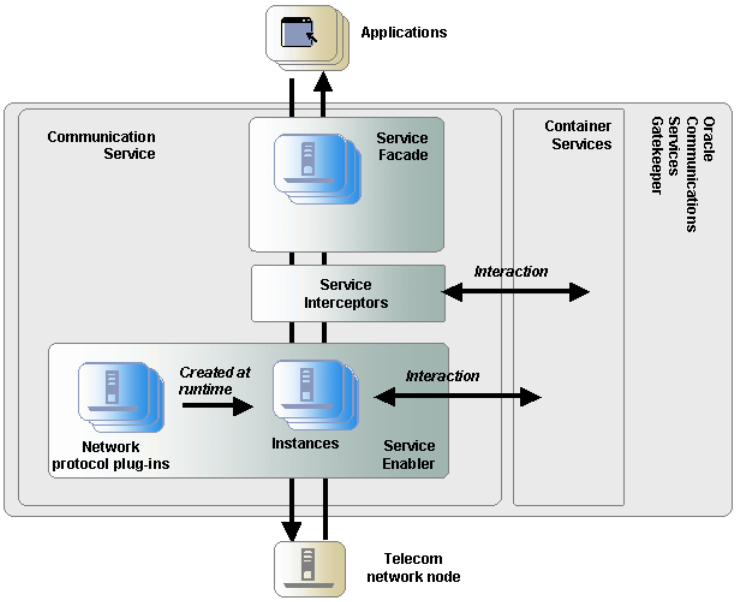
The communication services use container services, which are provided with the installation. Container services include Alarm, Event Data Record (EDR), Call Details Record (CDR), Policy Enforcement, Service Level Agreement (SLA) enforcement, Account, Event channel, and Trace services.

Requests between the network and applications can be intercepted by using service interceptors, which may allow, deny, or manipulate the request as necessary. When called upon to act on a request, service interceptors interact with container services as necessary to determine how to handle the request.

Service facades, service enablers, and service interceptors are deployable units in Services Gatekeeper.

Figure 1 illustrates how the Services Gatekeeper service components mediate the flow of requests between applications and the telecom network node.

Figure 1 Services Gatekeeper Components



Overview of Deployment Types

Services Gatekeeper supports the following types of deployments:

- Tiered deployments, which are suitable for large production environments
- SOA (Service-oriented architecture)/Tiered deployments, which are suitable for large production environments.
- Non-tiered deployments, which are suitable for test and development and small-scale production environments. There are two types of non-tiered deployments:
  - basic developer
  - basic high availability
- Geographically redundant deployments, which are suitable for large production environments in which provisioning and run-time processing data are replicated between sites

Table 1 provides a summary of the different deployment types.

Table 1 Summary of Deployments

Deployment Type	Provides	Characteristics
Tiered	Access and Network clusters	Targeted for medium and large deployments. Some high-availability aspects. High level of security. High level of scalability.

Table 1 (Cont.) Summary of Deployments

Deployment Type	Provides	Characteristics
Tiered/SOA	SOA functionality	Used if your implementation requires SOA functionality.
Non-tiered	Basic developer configuration	Targeted to extension and integration developers. No high-availability or redundancy aspects.
Non-tiered	Basic high-availability	Targeted for smaller deployments, testing, and development. Introduces support for high availability or redundancy.
Geographically redundant	Geographically separated sites with data synchronization	Each site has the characteristics of a tiered deployment. Adds geographical redundancy aspects that allow for disaster failover in the event of a site failure. Both sites are active; assumes site affinity from an application's point of view.

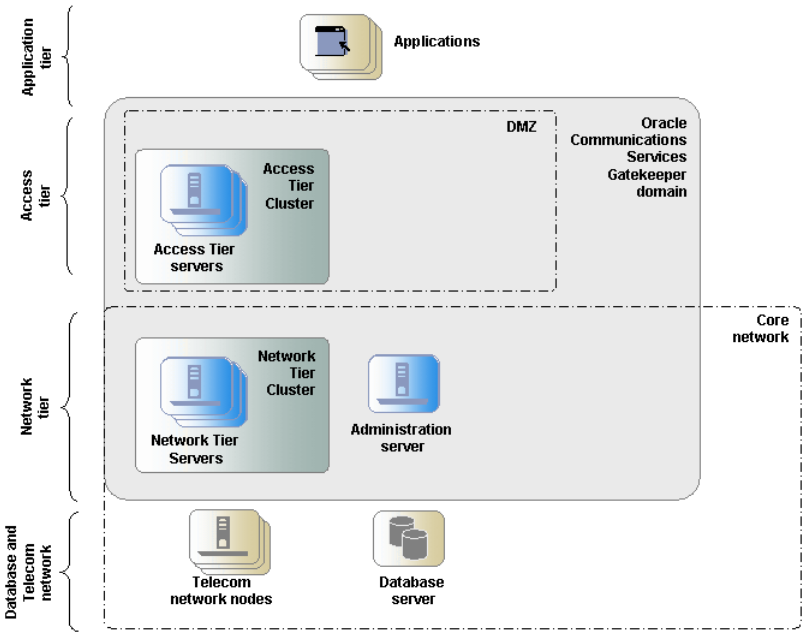
Tiered Deployments

A Services Gatekeeper deployment is normally divided into three tiers: the Access Tier, the Network Tier, and the Database Tier.

As Figure 2 shows, in-house and third-party applications that use Services Gatekeeper interact only with the Access Tier. The Network Tier interacts with the telecommunications network, the Access Tier, and other nodes such as Operation Support Systems (OSS) or Business Support Systems (BSS).

Service facades are deployed in the Access Tier nodes. Service enablers and container services are deployed in the Network Tier nodes.

Figure 2 Example of a Tiered Deployment



The tiering physically separates the servers, which enables carrier-grade scaling, security, and high availability.

Services Gatekeeper uses storage services that rely on an underlying database. For security reasons and in order to scale the database tier independently, the database is normally running on separate, dedicated nodes.

Physical Architecture

Each deployment consists of a number of nodes to ensure high availability and to provide redundancy and load balancing. The nodes are separated into a tiered architecture.

Production deployments of Services Gatekeeper are normally tiered into an Access Tier, a Network Tier, and a Database Tier.

The Access Tier is responsible for:

- Security
- SSL (secure sockets layer) termination
- XML serialization
- Termination of more latent WAN connections with applications

The Network Tier is responsible for:

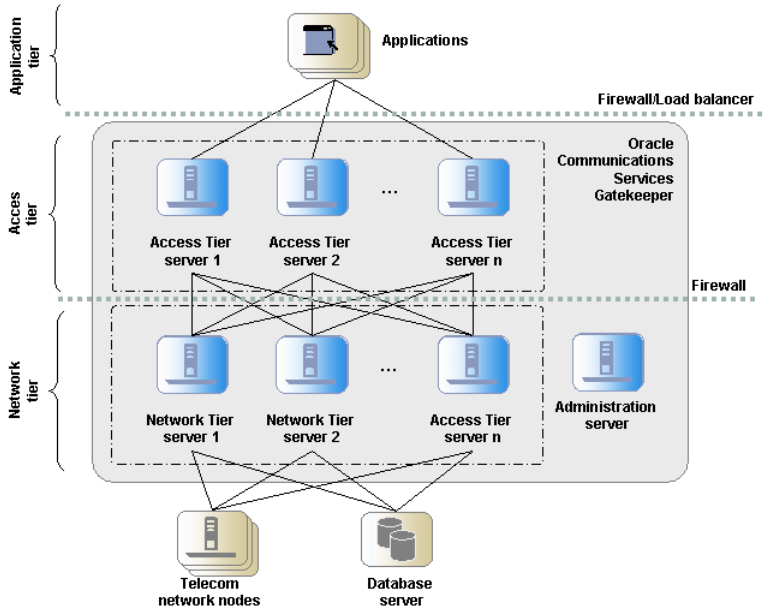
- Network protocol translation
- Generation of EDRs (event data records), CDRs (charge data records), and alarms
- Inter-node communications and state management

The Database Tier is a relational database management system (RDBMS) used to store configuration and provisioning data, as well as data generated as a result of

interaction with applications and network nodes. The RDBMS is abstracted by the Storage Service, a container service that provides the nodes with access to a shared cache. The Storage Service is deployed on nodes in the Network Tier.

Figure 3 shows an example of the tiered deployment of servers in a Services Gatekeeper installation. A firewall and a load balancer separates the Application Tier from the servers in the Access Tier. A second firewall regulates the traffic between the servers in the Access Tier and the servers at the Network Tier. All servers in the Network Tier are managed by the Services Gatekeeper Administration Server. The Managed Servers in the Network Tier can access the telecom network nodes and the database servers.

Figure 3 Servers in a Tiered Deployment



As seen in Figure 3, the tiers provide a separation at the network level, which allows for:

- Firewalls to be introduced between the tiers
- Different networks to be used for the different tiers

The servers are also physically separated within a tier.

Each tier consists of at least one cluster, with at least two server instances (nodes) 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.

Runtime Aspects

Nodes are grouped into one or more clusters in a deployment. With a few exceptions, all the same components are deployed on nodes within a cluster and these components are managed as one unit. There are a set of services where a component, a cluster singleton, is active only on one node within the cluster at any given point. In

case of node failure, the singleton service is automatically migrated to another node in the cluster.

Configuration settings for a deployed module can be per node or shared among the components deployed in the a cluster.

The clusters are grouped into a domain, with an Administration Server that normally does not process any traffic. The traffic-processing nodes are called Managed Servers.

There is normally a one-to-one relationship between Managed Servers and a physical servers. In some cases, several Managed Servers can run on the same physical server. If the CPU is powerful and has a lot of memory, performance can benefit from using a smaller heap size for a set of Managed Servers on a single physical server, rather than using one Managed Server per physical server. The disadvantage of this setup is mainly loss of redundancy and lower availability if a physical server fails.

Scalability

The Access Tier and the Network Tier scale independently of each other. New servers can be added at runtime, allowing you to scale the deployment horizontally.

The main responsibilities of the Access Tier are security, SSL termination, XML serialization and termination of WAN connections from applications. The main responsibilities of the Network Tier are to perform network protocol translation and protocol abstraction. This separation of responsibilities translates into two very different processing models.

Processing in the Access Tier is CPU-intensive, mainly concerned with XML to Java translations that have a short life span and generate numerous short-lived objects that trigger frequent garbage collection. The Access Tier does not maintain any state information. This behavior is consistent across communication services.

Processing in the Network Tier maintains state information and puts demands on data caching, inter-node communication, and processing logic.

The behavior of communication services varies with some of the services supporting relatively long-lived sessions. Call control sessions tend to have a significantly longer session lifetime than more data-centric sessions, such as messaging.

Some protocols, for example SMPP (short message peer-to-peer protocol), have more efficient data transfer sizes compared to XML-based protocols, for example MM7 (multimedia messaging service interface). This translates into different processing needs.

Sizing and configuring individual servers in the Network Tier depend on which communication services are used in the deployment and the estimated utilization ratio between them. Both the physical characteristics of the servers (such as internal memory, network cards and CPU speeds) and settings for the Java Virtual Machine, (such as heap size and other parameters that affect garbage collection) can be optimized for the different use cases.

In summary, the processing models determine how you optimize the physical hardware, the Java Virtual Machine, and the operating system. Tiering allows you to tune individual nodes in each tier for the different processing requirements.

Security

Services Gatekeeper provides extensive support for authentication, authorization, and accounting. In addition, the separation of physical tiers allows for network-level

security. This helps protect the network from attacks by fraudulent applications that use resources without paying for their usage and attacks designed to take resources out of service.

By using separate IP-network domains, one for the Access Tier and one for the Network Tier, you can apply different levels of network security. Applications are allowed to physically connect only to Access Tier servers, possibly fronted by a firewall, while the Network Tier servers only have access to the network domain where the telecommunication network nodes reside. In addition to this, the Access Tier servers are only allowed to connect to the Network Tier servers, possibly using a firewall between the tiers.

This topology puts the Access Tier servers in a demilitarized zone (DMZ) where out-of-network applications are only allowed access to the Access Tier servers. It also puts the Network Tier servers in a more strictly controlled domain, where the network elements they connect to are well known and the access is controlled by firewalls.

High Availability

From a high-availability perspective, tiered deployments are better than non-tiered deployments. Processing in the Access Tier is stateless and is characterized by negligible latency while processing in the Network Tier is stateful, involves more processing logic, and higher latency.

In a tiered deployment, the Access Tier adds a high-level load balancing function that is aware of the health of each Network Tier server and quickly removes the server that is out of service from the list of servers to load balance among. This means that fewer requests are affected if a fault occurs.

The Access Tier guarantees that requests toward the Network Tier are properly load balanced and that requests are not repeatedly sent to Network Tier servers that are out of service. Network Tier servers asynchronously update the Access Tier servers when they are back in service.

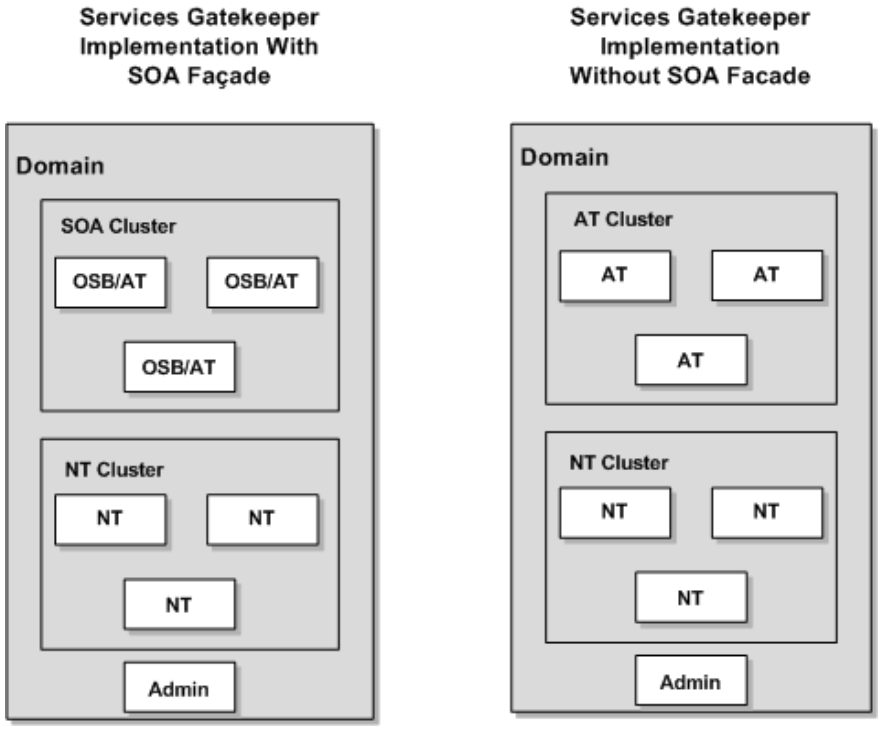
If a request from an Access Tier server targets a Network Tier server that has failed, the Access Tier server sends the request to another Network Tier server. The Storage Service provides reliable cluster-wide access to all state information kept by the Network Tier. As a result, any Network Tier server can process requests coming from any Access Tier node or network node. If a node fails, cluster singleton services are automatically migrated from the failed node to a healthy.

Tiered/SOA Deployments

SOA features require that you use a tiered deployment with the SOA co-located with your Access Tiers in an SOA cluster.

Figure 4 shows the difference between Services Gatekeeper implementations with and without SOA. Domains using SOA must be tiered and the SOA installation co-located with the Access Tier (AT) inside the domain. The rest of the domain structure, including NT clusters and Admin server, are denials to both implementations.

Figure 4 Services Gatekeeper Architecture with SOA Support



Non-tiered Deployments

Services Gatekeeper can be deployed in a non-tiered deployment by using one of the following configurations:

- Multiple node configuration: A non-tiered, multiple node configuration is targeted toward smaller production environments that have less strict scalability and high availability requirements.
- Single node configuration: A non-tiered single node configuration is targeted toward test and development environments.

Service facades, service enablers, container services, and service interceptors are deployed in all nodes in non-tiered deployments.

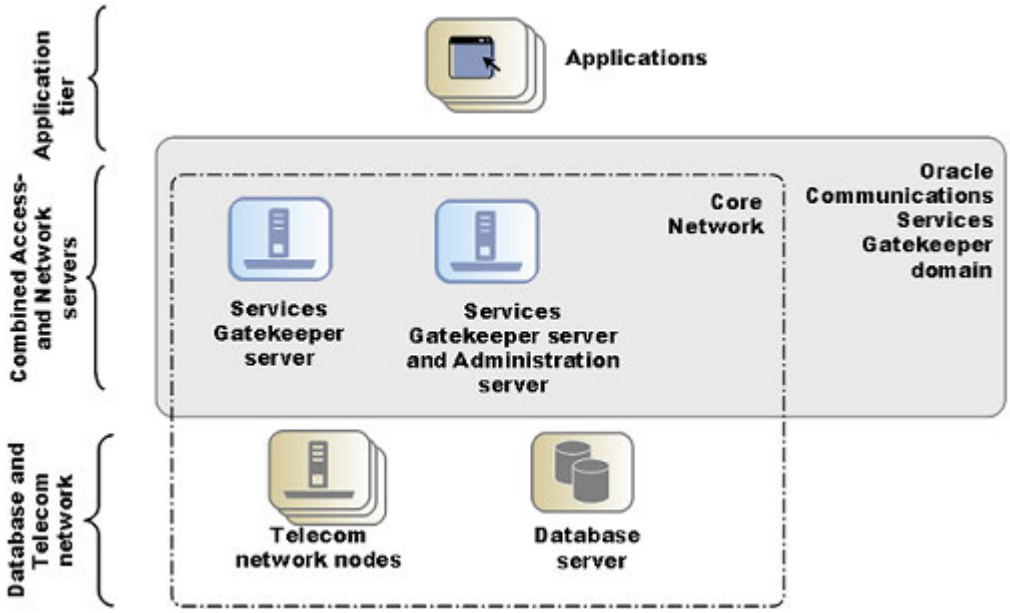
Using Non-tiered Deployments in Production Environments

You might use a non-tiered deployment in a production environment when security requirements and scalability requirements are irrelevant or minimal. An example of this is when Services Gatekeeper only serves applications hosted within the operators' domain and there is very restricted access to the IP network where Services Gatekeeper is deployed, and the integrity of the network is ensured by external mechanisms.

Scalability is compromised when you use a co-located Access and Network Tier, because the individual servers cannot be optimized according to their diverse processing models.

Figure 5 shows a deployment that has a cluster configuration. The Administration Server also processes traffic. The cluster can contain two or more servers.

Figure 5 Example of a Dual Server Non-tiered Deployment for Basic High Availability

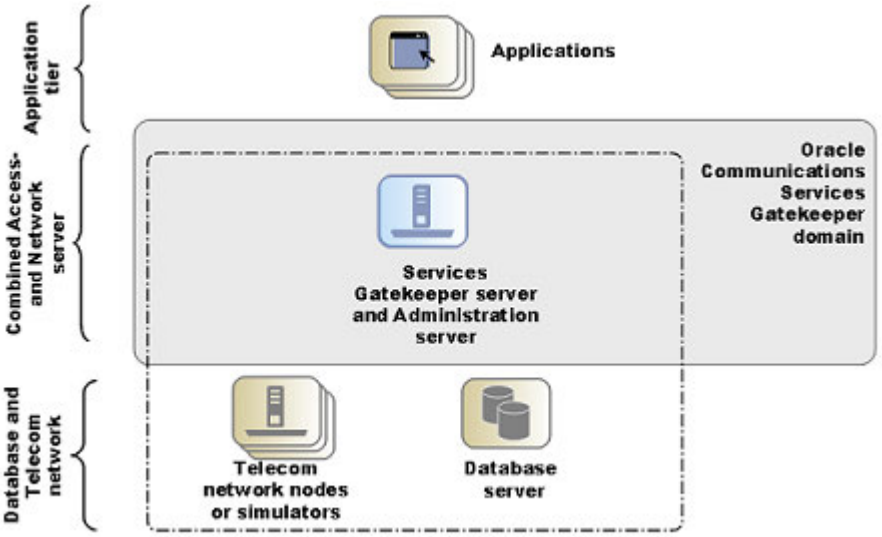


Test and Development Environments

When you use Services Gatekeeper for functional testing of extensions and integrations, there is no immediate need for a multi-server configuration. A single-server configuration can be used to simplify management and configuration for the developer or tester.

Although it is possible to run several servers on a single physical machine, the only reason to do so is to run initial high availability tests. System tests should be performed on a deployment with multiple physical servers.

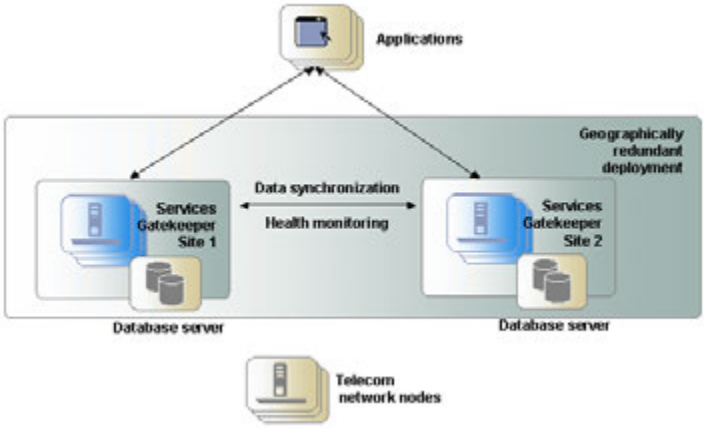
Figure 6 Example of a Single Node Non-tiered Deployment for Test and Development



Geographically Redundant Deployments

Geographically separated deployments are important for high availability. To prevent service failure in the face of catastrophic events such as natural disasters or massive system outages caused by power failures, you can deploy Services Gatekeeper at two geographically distant sites that are designated as site pairs. As Figure 7 shows, each site, which is a Services Gatekeeper domain, has another site as its peer. Application and service provider configuration information, including related SLAs and budget information, is replicated and enforced across sites.

Figure 7 Example of a Geographically Redundant Deployment



Geographically redundant sites are the active-active type, which means that both sites in a pair can be used to process traffic simultaneously. These deployments are connected by communication channels for data synchronization and health monitoring. The data synchronization replicates budget information between the site pairs to enforce Service Level Agreements (SLAs) accurately. Applications should have site affinity, but have the ability to fail over to the other site if necessary. Each site is managed and configured independently. Accounts and SLAs are replicated across sites. Typically, one Database Tier is used per site.

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 slave can be forced to synchronize to the geomaster. Any relevant configuration changes made to either site are written synchronously across the site pairs, so that a failure to write to one site causes the write to fail at the other site and an alarm is triggered.

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. This behavior applies only to global configuration changes.

For applications, geo-redundancy means that their traffic can continue to flow in the event of a catastrophic failure at an operator site. 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.

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 information 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 Services Gatekeeper or passed into Services Gatekeeper from the applications. Any state associated with a correlation identifier exists on only a single geographic site and is lost if an entire site goes down. 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.

As a result, conversations must be conducted and completed at their site of origin. If an application wants to maintain conversational state across sites, for example, to maintain a registration for network-triggered traffic, the application must register with each site individually.

Domain Templates for Deployment Types

The Services Gatekeeper installer includes a set of domain templates for each type of deployment that Services Gatekeeper supports.

Table 2 summarizes the deployment templates.

Table 2 Domain Templates for Deployments

Domain template Type	Description
Basic developer configuration with co-located Access and Network Tier	The Access and Network Tiers are co-located and there is no support for high-availability configurations. The server does not belong to a cluster but is tied to the domain. This deployment type is targeted for non-production environments where developers need access to Services Gatekeeper for functional testing of extensions and integrations.
Basic high-availability configuration	The Access and Network Tiers are co-located. The domain configuration templates support a setup with two servers in the same tier. One of the servers can also serve as the WebLogic Administration Server. The servers do not belong to a cluster but are tied to the domain. Database replication is not automatically provided, but must to be configured at the database level. This deployment type is targeted toward non-production environments where developers need access to Services Gatekeeper for non-functional testing such as basic high-availability testing of extensions and integrations. It is also targeted for basic, entry-level production environments that have limited requirements for security, because it does not support a DMZ separated by a firewall. It also provides the very minimal redundancy because it supports only two-server setups.

Table 2 (Cont.) Domain Templates for Deployments

Domain template Type	Description
Access and Network Tier clusters	The Access and Network Tiers are separated in clusters. The domain configuration templates supports multiple servers in each cluster. A separate server has the role of a WebLogic Administration Server. This server does not process traffic requests. High availability toward the database is not supported automatically. Redundancy toward the database is up to the database deployment. This deployment type is targeted toward production environments and supports deployments with a DMZ between the Access and Network Tier.
Access and Network Tier clusters with Oracle RAC database	This configuration has all the properties of the Access and Network Tier cluster setup and adds high availability and redundancy toward the database. This setup leverages from the failover and redundancy characteristics of Oracle Real Application Cluster (RAC). This deployment type is targeted towards production environments and supports deployments with a DMZ between the Access and Network Tier.

About Deploying the Database

The deployment architecture strongly favors deploying the database in a separate tier. The database should be deployed on dedicated servers for both security and performance reasons. Backup and other data-intensive operations should not load traffic-processing servers. Database administrators should be granted exclusive privileges to log on and perform SQL operations. Configuration of Services Gatekeeper components should be performed by using the Services Gatekeeper management interfaces and should not be performed directly at the database level.

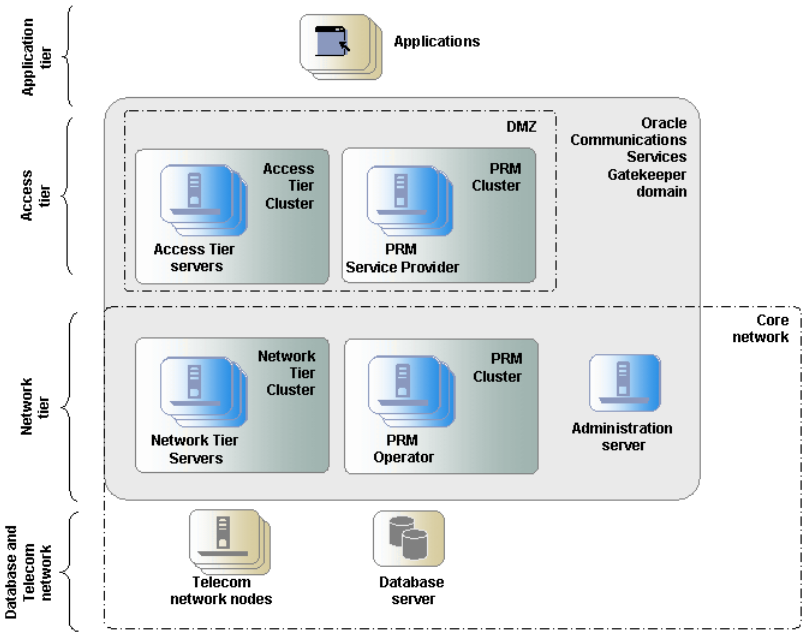
Oracle recommends that you use an Oracle database, where the instance is based on the transaction processing template, runs in dedicated server mode, and uses automatic store management.

About Deploying Partner Relationship Management Modules

Oracle recommends that you deploy the Partner Relationship Management (PRM) modules in a separate cluster in the domain. The PRM servers should not be co-located with Access Tier servers or Network Tier servers, because PRM processing impacts the performance of processing traffic requests.

There are two views to PRM: the service provider view and the operator view. Each view can be deployed separately. A portal application can benefit from being deployed in two parts: the service provider view deployed in the DMZ and the operator view deployed inside the secure network of the operator, not accessible from the Internet.

Figure 8 Example of a PRM Deployment



About Deployment Administration

All management, configuration, and provisioning operations at the JEE level are performed by using the Administration Server and are propagated to the relevant Managed Servers when the servers are deployed in clusters. Operations includes starting and stopping Managed Servers and deploying and undeploying Services Gatekeeper container services and communication services.

Container Services and Communication Services using the Services Gatekeeper MBeans can be performed on any of the servers when the configured attribute is shared among all instances in the Network Tier cluster. When the configuration attribute is local for the server, the attribute must be configured on the individual server.

Services Gatekeeper components can be configured, managed, and provisioned by using a Web-based administration GUI, interactive-text mode, scripting, and JMX.

For a complete list of Services Gatekeeper administration tasks and instructions, see *Oracle Communications Services Gatekeeper System Administrator's Guide*.

About Integrating Services Gatekeeper with Service Controller

You can integrate Services Gatekeeper with Oracle Communications Converged Application Server, Service Controller edition (Service Controller) if your implementation requires its service orchestration and protocol mediation capabilities.

A Service Controller-Service Gatekeeper integration must communicate using SIP traffic, and Services Gatekeeper must then translate the SIP traffic into SS7 format. Consequently, these are the network-facing communication services that can take advantage of the integration:

- Parlay X 2.1 Audio Call/SIP

- Parlay X 2.1 Call Notification/SIP
- Parlay X 2.1 Presence/SIP
- Parlay X 2.1 Third Party Call/SIP
- RESTful Third party Call
- RESTful Call Notification
- RESTful Audio Call
- RESTful Presence

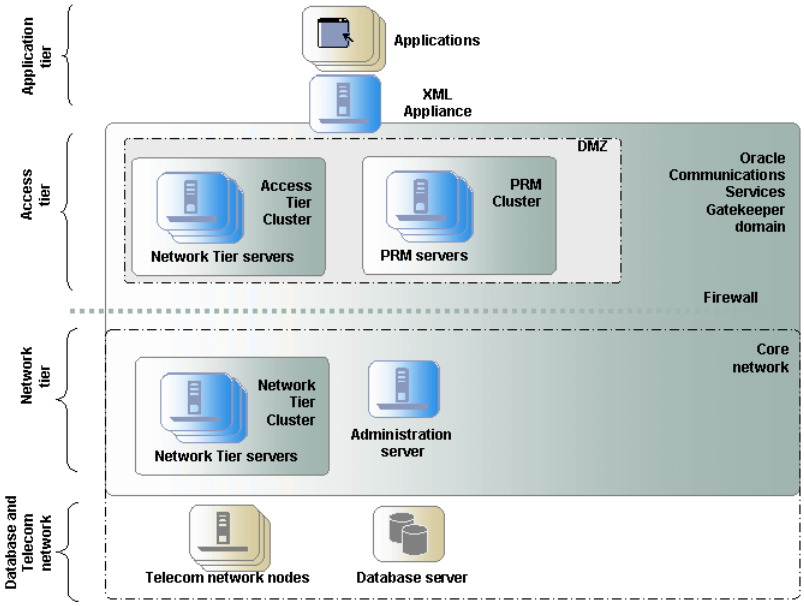
For details on these communication services see the *Oracle Communications Services Gatekeeper Communication Service Guide* and the *Oracle Communications Services Gatekeeper RESTful Application Developer's Guide*.

About XML Appliances

XML appliances provide complement functionality provided by the Access Tier. They provide:

- XML acceleration using hardware in the form of ASIC circuits optimized for XML parsing, XPATH processing, and XSLT transformation and validation.
- Security applications, especially for:
  - XML screening for intrusion detection, traffic monitoring, and content filtering.
  - Firewall and Virtual Private Network applications for authentication, Web Services Security compliance and Single Sign-On.
  - Network gateway applications such as routing, traffic management, and protocol translation.

Figure 9 Example of a Deployment with XML Appliances



XML appliances in the form of firewalls are recommended in all deployments and are important between the Access Tier and the Internet. They facilitate setting up secure channels between applications and Services Gatekeeper.

XML Appliances do not provide high availability retries, health checks, or automatic synchronization of in-production upgrades.

The communication protocol between the Access and Network Tier is not XML. The Network Tier accepts and performs Java RMI calls.

Adding XML appliance in front of a Services Gatekeeper deployment adds another layer of latency.

Firewalls can be introduced between the Access Tier and the Network Tier. The tiered deployment model supported by Services Gatekeeper makes it very suitable for this.

To ensure network integrity, the Access Tier provides a set of carrier-grade security mechanisms that include:

- Web Services Security standards
- Message-level security, encryption, and trust
- Transport-level security
- Authentication, authorization and accounting (AAA)

All of these mechanisms are direct results of using WebLogic Server as a container. The authentication parts must be compliant with the account model used in Services Gatekeeper, unless double authentication procedures are performed, one procedure for the above and one for the Services Gatekeeper account.

For access control, Services Gatekeeper provides a set of enforcement rules, including time of day, day of week, and is capable of taking historical data into account. Examples of historical data are aggregated number of requests over a number of days or peak request rates expressed in milliseconds.

Services Gatekeeper deployments can benefit from the use of existing XML appliances, especially for the following use cases:

- Firewalls, both for fronting the deployment and for separating the Access and Network Tier.
- Load balancers, to balance the load between servers in the Access Tier.
- XML Schema validation (including message size, element size, and string lengths) for additional security and to off-load Services Gatekeeper message-validation processing.
- SSL termination points, to off-load the message-level and transport-level security processing from the Access Tier.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Oracle Communications Services Gatekeeper Deployment Guide, Release 5.1
E37527-01