Oracle® Communications Service Controller

Concepts Guide

Release 6.2

F18707-02

April 2020

ORACLE®

Oracle Communications Service Controller Concepts Guide, Release 6.2

F18707-02

Contents

Preface

This document provides an overview of Oracle Communications Service Controller.

Audience

This document is intended for anyone who installs, configures, or administers Service Controller. It should also be read by all users who want to understand key concepts of Service Controller.

It is assumed the reader is familiar with telecommunications network architectures and technologies such as IP Multimedia Subsystem (IMS) and SS7-based networks, and with telecommunications network protocols, especially the Session Initiation Protocol (SIP), Diameter, and SS7-based protocols.

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 that have purchased support 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.

1

Service Controller Overview

This chapter provides an overview of the Oracle Communications Service Controller product line.

About Service Controller

Service Controller is a carrier-class platform supporting an open architecture that makes it easy to integrate with new technologies and networks.

Service Controller provides real-time service orchestration and protocol mediation capabilities. With a versatile orchestration engine and portfolio of standard interfaces, the product provides a gradual migration path from legacy infrastructure to next generation platforms. It enables CSPs to launch innovative new services that blend legacy and Internet Protocol (IP) networks.

With Service Controller, legacy services running on Service Control Points (SCPs) and Session Initiation Protocol (SIP) services running on application servers, gain access to and control sessions across legacy networks and SIP networks. Service Controller decouples services from network infrastructure, enabling services and networks to seamlessly converge.

Service Controller enables any network to integrate with any other network. With a full suite of SS7 interfaces, Service Controller translates any Intelligent Network (IN) protocol to any other protocol, including SIP and other variants of the original IN protocol. IP Multimedia Subsystem (IMS) assets and IN assets are brought together through the orchestration engine to build cross-domain services.

Included with Service Controller is a SIP interface for implementation of SIP-based services. Services using this interface control sessions in both the IMS and legacy domains, provided with an extensive set of capabilities available in the network, ranging from basic session redirection and disconnection to announcement playing.

Services can be implemented and executed on any standard SIP application server, such as Oracle Communications Converged Application Server. See *Service Controller SIP Developer's Guide for GSM* for more information.

2

Service Controller Tier Architecture and Administration Model

This chapter describes the common Oracle Communications Service Controller deployment concepts.

Understanding the Signaling and Processing Tiers

Service Controller deployments separate processing functionality from signaling functionality, creating two tiers:

- Signaling tier

  The signaling tier executes protocol stacks that provide industry-standard interfaces between the processing tier and networks. All of the traffic exchanged between Service Controller processes in the processing tier and network entities running on external systems is routed in and out through the signaling tier.

  The processes running in the signaling tier are called Signaling Server Units (SSUs). Network entities connect to Service Controller through SSUs. See "About Signaling Server Units in the Signaling Tier" for more information.

- Processing tier

  The processing tier executes the main functions for which you deploy Service Controller. For example, in the a Service Controller deployment the processing tier executes mediation and orchestration functions.

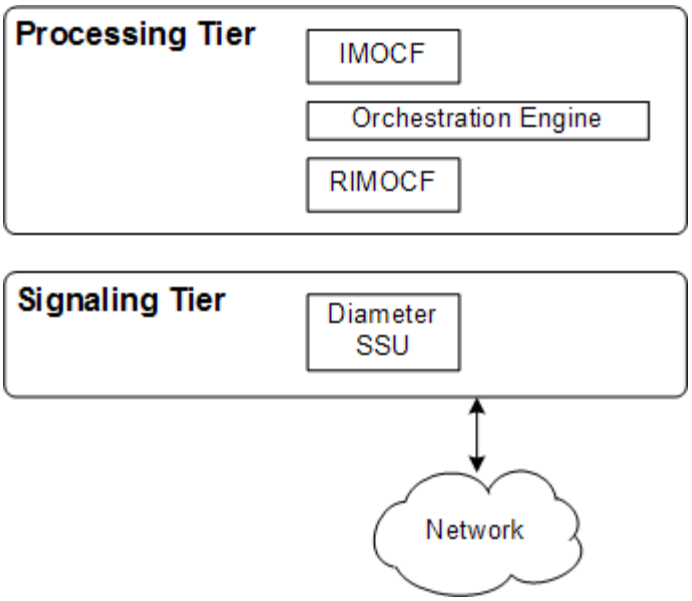
Each of the tiers, both the signaling and processing, is implemented using one or more servers. Servers are named after the tier they belong to, that is processing servers and signaling servers. The tiers are scalable; You can add as many servers as you need to each of the tiers.

SSUs running in the signaling tier are stateless, but components running in the processing tier are stateful. Components retrieve and store session state in an in-memory storage. Session state is maintained and distributed across the servers in the processing tier.

Normally a production deployment includes at least four servers, two in each tier, to support service availability and service continuity. See "Reliability" for more information.

Figure 2–1 shows the components running in the signaling tier and processing tier when deploying Service Controller. In the signaling tier, SSUs provide connectivity to Online Charging Systems and other relevant network entities through the Diameter protocols. In the processing tier, modules (IMOCF, RIMOCF) provide the mediation functionality, and the Orchestration Engine provide the orchestration functionality.

Figure 2–1 Components in the Signaling Tier and Processing Tier



About Signaling Server Units in the Signaling Tier

Signaling Server Units (SSUs) are processes that run on servers in the signaling tier. SSUs provide Service Controller with industry-standard interfaces to the network. All the traffic exchanged between Service Controller and entities in the network, is routed in and out through the SSUs. SSUs route outbound traffic from modules in the processing tier to entities in the network, and inbound traffic from entities in the network to modules in the processing tier.

You use the Administration Console to configure SSUs. See *Service Controller Signaling Server Units Configuration Guide* for reference information about configuring the different types of SSUs.

Inbound Routing

All messages that are targeted at Service Controller arrive through SSUs. When messages from the network arrive to the SSU, the SSU routes the messages to modules in the processing tier. The SSU performs a routing decision when the first message of a new session arrives. Subsequent messages in the same session are routed to the same module.

The routing decision is based on inbound routing rules you configure in the SSU. In the routing rule, the destination module is specified by the module name and module type. If a system includes more than one processing domain, then the destination module is specified also by the name of the processing domain where the module is deployed. For example: imocf.IMOCF@oracle-domain.

In the routing rule you also specify criteria for selecting each destination module. The criteria is based on values of parameters inside the initial message. The criteria depends on the type of SSU and the protocol the SSU supports.

When the SSU routes a message to a destination domain, any server in the domain can process the message equally.

Outbound Routing

All messages that are sent from modules in the processing tier to external entities in the network, go out through SSUs. When messages from modules arrive to the SSU, the SSU routes the messages to entities in the network.

You use the SSU configuration screens to configure destination entities in the network. The SSU monitors the state and availability of every destination entity. For each destination entity you also define an alias. If a module in the processing tier needs to interact with a particular entity in the network, you configure the address of the destination entity using the alias you defined in the SSU for the particular destination entity.

When the SSU routes messages from modules to the network, the SSU resolves the alias to a real network address. If you define the same alias for two or more entities in the network, the SSU will balance the load among the available network entities, based on their status.

Types of SSUs

Service Controller supports the following types of SSUs:

- SSU SS7 for SIGTRAN, which provides access to legacy SS7 network through M3UA protocols.
- SSU SIP, which provides access to SIP-based networks.
- SSU Diameter, which provides access to entities in the network through various Diameter application protocols, such as Ro.
- SSU SMPP, which provides access to Short Message System Centers (SMSC) through the Short Message Peer-to-Peer (SMPP) protocol.
- SSU WS, which provides access to network entities through SOAP and REST-based communication.

About Processing Modules in the Processing Tier

There are various types of modules that you can deploy in the processing tier, and they all differ by the functionality they provide. After you install Service Controller, you deploy the kind of processing modules required for your solution.

For information about the specific processing modules available in the processing tier, refer to the implementation guide of the Service Controller product that you deploy.

Service Controller Administration Model

The Service Controller administration is based on the concept of *domains*. A domain is a logically related group of servers.

In a typical Service Controller deployment you administer the group of logically related signaling servers in one domain, the signaling domain, and the group of logically related processing servers in another domain, the processing domain. See "Domain Types" for more information about the different types of domains.

Servers in the domain are referred to as *managed servers*.

About Managed Servers

A server in the domain is called *managed server*.

Each managed server runs on its own Java Virtual Machine (JVM). All servers in the domain have the same Open Services Gateway Initiative (OSGi) software bundles installed and started, so they all provide the same functionality and can equally provide services.

Managed servers have read-only access to the domain configuration directory. For more information about the domain configuration directory, see "About the Domain Configuration Directory". At startup, the servers load the software bundles and the required configuration from the domain configuration directory.

Managed servers can be added and removed from the domain without service interruption while the system is running.

About the Domain Configuration Directory

Each domain has one associated domain configuration which is stored in the domain configuration directory. The domain configuration directory contains:

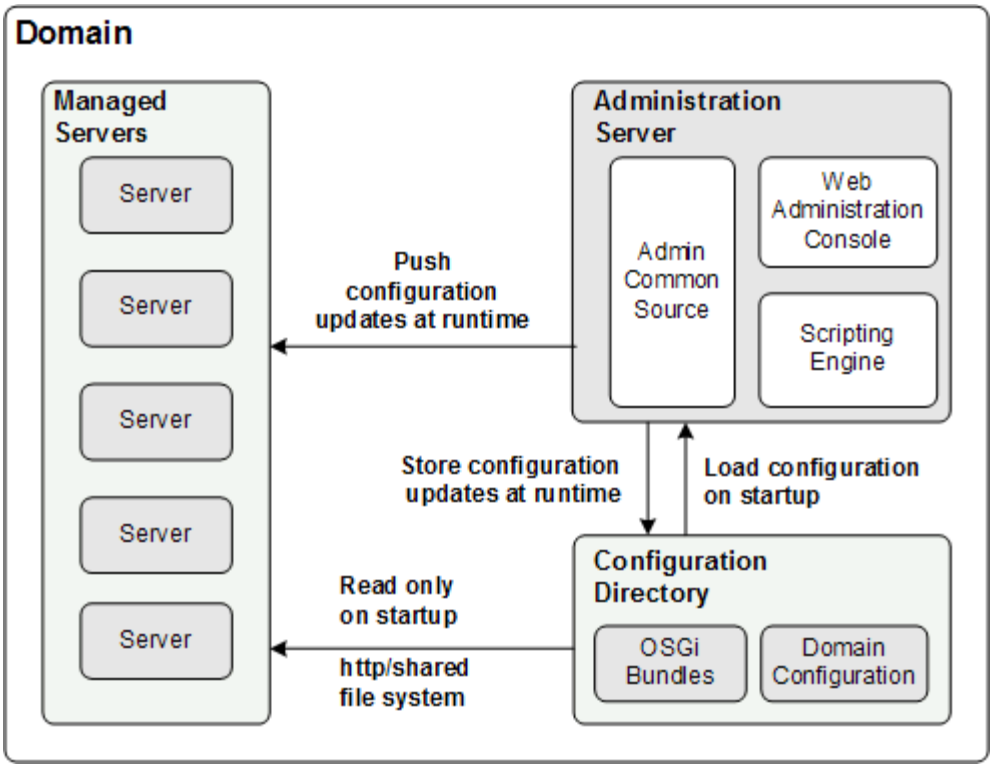
- The configuration details stored in XML files for all the managed servers in the domain.
- The list of OSGi software bundles that the managed servers run. This list defines the functional role of the managed servers.

When a managed server starts, it retrieves configuration data from the domain configuration directory and loads it into memory. Individual servers do not store configuration locally except for the initial configuration and the security-related details they need to enable access to the domain configuration directory.

Changes in the domain configuration directory are managed by the Administration Server. The Administration Server saves all configuration updates to the domain configuration directory during runtime.

Figure 2–2 shows how managed servers obtain the data in the domain configuration directory.

Figure 2–2 Management of the Configuration Data in the Domain



Managed servers access the domain configuration directory using either of these data sharing methods:

- Shared network file system
- Domain Configuration Web server that connects to the managed servers using HTTP or HTTPS.

> **Note:** HTTPS should always be used in a production environment.

About the Administration Server

The Administration Server enables you to manage the domain servers, the OSGi software bundles installed and deployed in the domain, and the data stored in the domain configuration directory. Oracle recommends using a dedicated computer for the Administration Server.

You can access the Administration Server using these clients:

- Administrator Console Web-based client:

  Web access enables administrators to configure the domain from any computer with a Web browser and network access to the Administration Server.

- JConsole or Scripting Engine:

  If you want, you can interact programmatically with the Administration Server by using JMX configuration MBeans. Typically, working with MBeans involves integrating Service Controller with a JMX-enabled network management system.

  Scripts can be used if you need to repeat lengthy and complicated configuration changes. The scripting engine is a shell script that accepts an XML file argument.

The XML file defines operations and attributes on Administration Server configuration MBeans.

Domain Types

The domain type reflects which sets of software bundles are running on the managed servers and which functional tier they implement.

There are three domain types:

- Signaling domain:

  Managed servers in the signaling domain run the software bundles associated with the signaling tier. These components include the various signaling server units (SSUs) that enable network connectivity. Servers in the signaling domain are often referred to as signaling servers.

- Processing domain:

  Managed servers in the processing domain run the software bundle associated with the processing tier. These components include the modules, an Orchestration Engine (OE), applications, and mediators that enable traffic processing and mediation functions. Servers in the processing domain are often referred to as processing servers.

- Unified domain:

  This domain combines the processing and signaling tier functions. Managed servers in the unified domain run the bundles associated with both the signaling tier and processing tier.

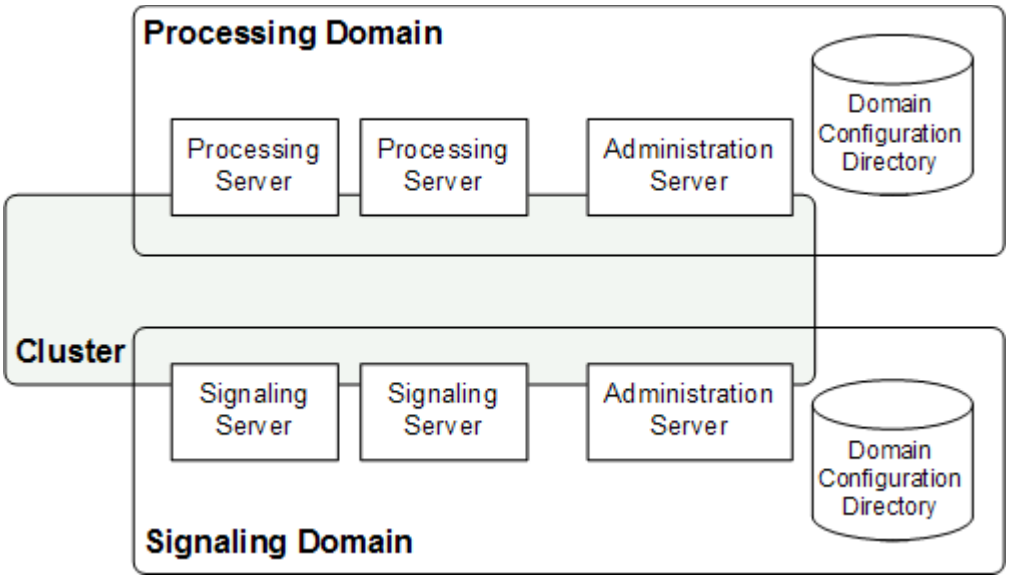
Cluster Architecture

Service Controller deployments are clustered, using Oracle Coherence. Coherence is a data management system for application objects that are shared across multiple servers. Coherence provides replicated and distributed data management and caching services on top of a reliable, scalable peer-to-peer clustering protocol. Coherence has no single points of failure; it automatically and transparently fails over and redistributes its clustered data management services when a server becomes inoperative or is disconnected from the network.

When a new server is added, or when a failed server is restarted, it automatically joins the cluster and Coherence fails back services to it, transparently redistributing the cluster load. When you deploy Service Controller, you do not need specific Coherence knowledge, nevertheless, for more information about Coherence see the Coherence documentation.

All servers of a Service Controller deployment belong to one cluster and equally act as the cluster members. Figure 2–3 shows a typical deployment with one processing domain and one signaling domain. Each domain includes two servers and an Administration Server. The figure shows how all servers and Administration Servers belong to the same cluster.

Figure 2–3 A Clustered Deployment



The cluster provides the following services:

- **Internal communication:** Internal communication among servers within a domain and across domain boundaries is based on the cluster services. If you choose a cluster multicast address for the exchange of information among servers, you need to configure all servers to use the same multicast address. You configure the multicast address for all servers in a domain when you create a domain. In a deployment with multiple domains, you configure the same multicast address for all the domains.
- **Distributed cache:** The distributed cache allows servers to distribute (partition) data across the domain so that each piece of data in the cache is stored by only one server. Each piece of data is backed up by one or more other servers to ensure that there is no data loss.
- **Redundancy:** All domain servers are known and are members of the cluster, therefore, the death of any server does not cause any loss of data.
- **Failover:** The death of a server is automatically and quickly detected, therefore failover occurs very rapidly, and more importantly, it occurs transparently.
- **Load balancing:** Since all servers are known, it is possible to load balance responsibilities across the domain. Load balancing automatically occurs to respond to new servers joining the domain, or existing servers leaving the domain.

Reliability

Service Controller is highly reliable in terms of failure probability and failure frequency. Service Controller supports:

- Service Availability

  Ensures that Service Controller is constantly available to handle new sessions. The domain model, of multiple redundant servers ready to process new sessions at any time, ensures that whenever a new session arrives, at least one server is available to process it.

- Service Continuity

Ensures that Service Controller is constantly available to handle existing sessions. Having multiple servers in a system does not for itself assure that existing calls will continue if one of the servers fail. The reason is that components in the processing tier are stateful and maintain session information. If one server fails there has to be a way to replicate the session state to another server that can take over the session processing.

Service Controller components in the processing domain maintain session information. They retrieve and store session state in an in-memory storage which is based on the cluster distributed cache service. Session state is maintained and distributed across the domain servers. On server failure, functioning servers continue to retrieve and process all messages, including those stored in the in-memory state of the failed server. Therefore, if a server fails, another server continues to handle existing calls, providing service continuity.

Service Controller uses Coherence for its in-memory data-grid. See the Coherence documentation for more information.

Service availability is an inherent capability of the domains model. However, if you want your deployment to support service continuity, you need to explicitly select it during installation. For more information see the section about Service Mode in *Service Controller Installation Guide*.

Scalability

Scalability is the ability of a system to provide throughput in proportion to, and limited only by, available hardware resources. A scalable system is one that can handle increasing numbers of requests without adversely affecting response time and throughput.

The growth of computational power within one operating environment is called vertical scaling. Horizontal scaling is leveraging multiple systems to work together on a common problem in parallel.

Service Broker scales both vertically and horizontally. Scaling options differ according to whether you are scaling the processing tier or the signaling tier.

See the section about scaling the Service Controller deployment in *Service Controller Installation Guide* for more information.

Open Services Gateway Initiative (OSGi) Framework

Service Broker is implemented using the Open Services Gateway initiative (OSGi) framework. Service Broker components in both the signaling and processing tiers are packaged and deployed as OSGi bundles.

You can install, start, stop, update and uninstall Service Controller bundles without rebooting servers in the signaling domain and processing domain.

The use of OSGi simplifies the Service Broker upgrade procedure and reduces its memory consumption.

For more information about OSGi, see the OSGi Alliance Web site:

http://www.osgi.org/Technology/HomePage