eServices Multitenancy and Load Balancing Guide

Genesys Engage Digital (eServices) 8.1.4

12/29/2021

Table of Contents

eServices Multi-tenancy and Load Balancing Guide

This guide covers:

- Multi-tenancy—The capability of maintaining a pool of resources and controlling access to it
- Load balancing—Providing multiple instances of certain servers for greater scalability and availability of service

Multi-Tenancy

Multi-tenancy is the capability of maintaining a pool of resources and controlling access to it by more than one tenant or business instance. This section covers:

- Configuring
- Limitations

Configuration

The following eServices applications can be shared across tenants: Chat Server, Interaction Server, SMS Server, Social Messaging Server, Universal Contact Server (UCS), Web API Server, and Classification Server. For all other eServices applications you must deploy one instance per tenant. In a multi-tenant environment, each eServices application must have one Tenant specified on its `Tenants` tab, with the following exceptions:

- UCS, Chat Server, Classification Server, SMS Server, and Interaction Server can have more than one Tenant specified.
- Web API Server can have more than one Tenant specified if the server itself is an object of load balancing.

This means that clients of Web API Server must include a `tenant` parameter in their requests, even in single-tenant environments, and even if the client itself is single-tenant.

> **Important**
>
> Changes to the tenant specification of SMS Server and Social Messaging Server do not take place dynamically—you must restart these servers for changes to take effect.
>
> The same applies to Interaction Server prior to release 8.1.2. However, in Interaction Server 8.1.2 and later, changes to the tenant specification take effect immediately.

Interaction Server

Since release 7.2, Interaction Server has had two possible Application types, `Interaction Server` and `T-Server`. With type `Interaction Server` it does not require an associated multimedia switch. Clients of Interaction Server should not expect it to specify any switch.

> **Important**
>
> Support of multi-tenancy requires the use of the `Interaction Server` application type.

For backward compatibility with release 7.1, clients are nevertheless able to associate Interaction Server with a multimedia switch, as follows: If a Tenant that is specified by Interaction Server contains a multimedia switch, clients associate Interaction Server with this switch. This works only if the Tenant contains exactly one multimedia switch.

There are also other consequences of this difference in application type.

Integrated Capture Points

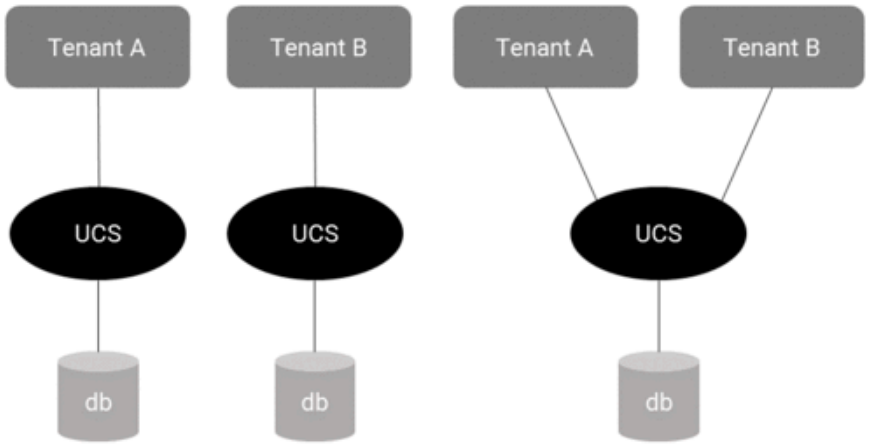
Capture Points support multi-tenancy by mapping each particular interaction to a tenant based on configured attributes. Refer to eServices Integrated Capture Points Guide for more information about Capture Points functionality in Interaction Server.

Limitations

Observe the following limitations:

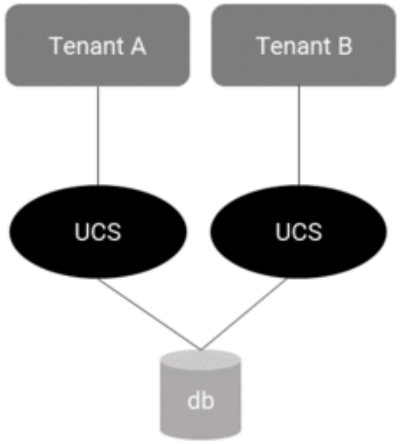
- Deploy at most one multimedia switch per tenant.
- Deploy at most one Interaction Server per tenant.
- Deploy at most one UCS per tenant.
- Deploy at most one UCS database per tenant. Databases can be shared between tenants. The figure below shows possible ways of arranging multiple tenants, Universal Contact Servers, and databases.

Supported Architectures



Unsupported Architecture

The architecture below remains untested and unsupported.


Tenant A
Tenant B
UCS
UCS
db

Load Balancing and Backup Configuration

Load balancing provides greater scalability and availability of service by providing multiple instances of certain eServices servers. The redundancy provided by load balancing helps prevent loss of data.

Load balancing can take place within a tenant or across tenants. There are three types of load balancing:

- Web API Server can balance among multiple instances of the following servers within a single tenant or across tenants:
  - Chat Server
  - E-mail Server
  - Interaction Server
  - Stat Server
  - UCS

    Web API Server does this using the Load-Balancing API. It keeps track of available server instances through Solution Control Server (SCS).
- Web API Server can balance among multiple instances of its own type (multiple Web API Servers).
- Interaction Server can balance, within a single tenant only, among multiple instances of Classification Server and E-mail Server. It can also balance among multiple instances of Universal Routing Server (URS) as long as all instances have the same strategy loaded. See the Interaction Server topic for details.

Web API Server

Web API Server uses Solution Control Server (SCS) to monitor the state of all servers. Load balancing that involves Web API Server includes:

- Load balancing between instances of Web API Server.
- Load balancing between instances of the following servers: Callback Server, Chat Server, E-mail Server, Interaction Server, Stat Server, UCS.

Load-balancing components react to:

- Changes in the configuration; specifically, the connection settings of Web API Server and the status (enabled or disabled) of relevant applications.
- Application states (running or stopped) as reported by SCS.

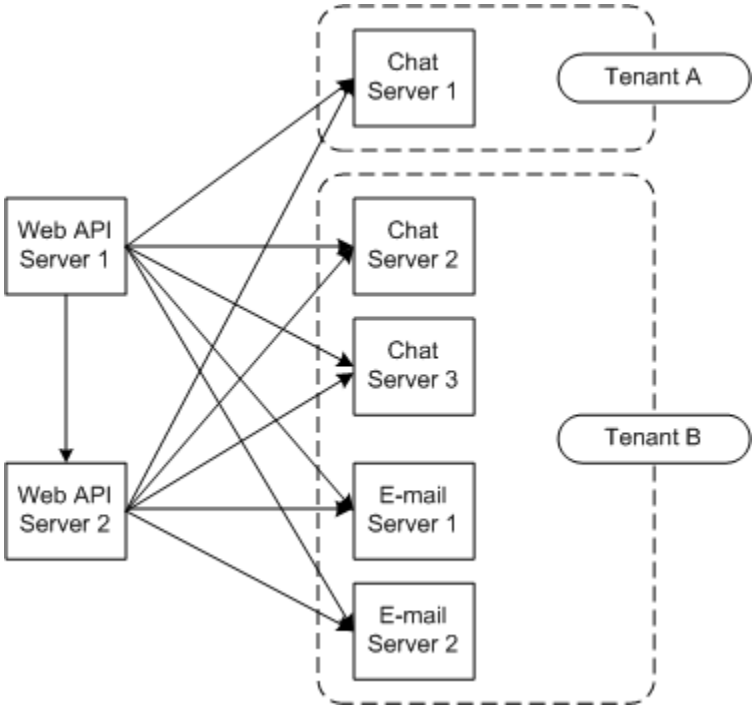
Load balancing performs the following actions:

- For all instances of servers that are listed on Web API Server's `Connections` tab:
  1. The instance reports its status as RUNNING to SCS when it starts. The instance is then ready to process interactions.
  2. The load balancer then marks the server's status as RUNNING.
  3. A web application can now use the service provided by that instance.
- If the server's status changes to Service Unavailable:
  - It is excluded from the list of available servers and cannot be given in response to a web application's request.
  - If the server's status changes again to Started (Service Available), it returns to the list of available servers. In the case of Chat Server, it can continue to handle an online session that was established before the status changed to Service Unavailable.
- If the server shuts down:
  - It is excluded from the list of available servers and cannot be given in response to a web application's request.
  - All interactions that it was handling are either lost (in the case of Chat Server) or handled by other instances of the same type of server (E-mail Server).
- If the server is disabled in the Configuration Layer:
  - It is likewise excluded from the list of available servers and cannot be given in response to a web application's request.
  - None of the clients that are already working with this instance of the server are affected.
  - As soon as all interactions that the server is handling end, the server can be shut down.
- If the server is enabled, it returns to the list of available servers.

Load-Balancing Configuration for Web API Server

In the example eServices configuration shown in below, there are two instances of Web API Server, two of E-mail Server, and three of Chat Server, with the Chat Server instances divided between two tenants. All application servers are available to both Web API Servers.

> **Important**
>
> To enable cross-Tenant load balancing, you must add the Tenants to the `Tenants` tab of the Web API Server that you want to serve as the point of load balancing.



Load Balancing Configuration

To simplify the overall configuration, you can use application objects of type `Application Cluster` to group available servers in the configuration. An Application Cluster is a configuration object that stores connection specifications. The eServices configuration wizard offers the opportunity to create Application Clusters. If you did not create an Application Cluster while running the wizard, you can add one manually, using the following procedure.

Configuring an Application Cluster

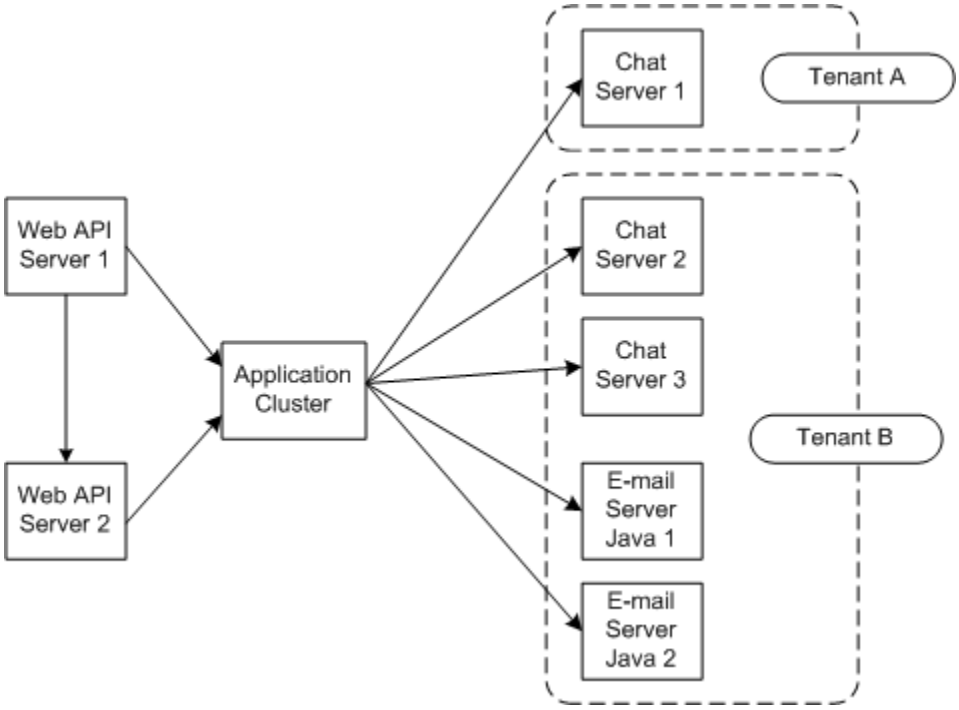
Start

1. In Configuration Manager, locate the template `ApplicationCluster_<version-number>.apd` in the `templates` directory of your product CD and import it.
2. Use the template to create a new Application Cluster.
3. On the `Connections` tab, add connections to the servers that Web API Server will balance.
4. On the `Server` and `Start Info` tabs, enter arbitrary characters in the empty fields. Web API Server ignores this information, but there must be something in these fields for you to be able to save the Application.
5. In your Web API Server Application object or objects, add a connection to the Application Cluster.

End

Application Clusters are transparent to load-balancing components, and the configuration that uses connections through Application Clusters is equivalent to a configuration that uses direct connections between servers.

The configuration shown in the previous section is reconfigured using an Application Cluster in the figure below.



Load Balancing with Application Cluster

Load-Balancing API

To use eServices 8.1 load-balancing capabilities, use the Load-Balancing API in the web application. The Load-Balancing API provides the following functionality:

- It can select a particular server instance from a set of instances of the specified server type.

- Upon the first request to a server instance, it can create an alias for the selected server instance and store it for future use.
- It can use the alias to obtain connection parameters (host name and port) of the server instance.
- It has access to configuration information.

The eServices 8.1 simple samples demonstrate how to use the eServices Load-Balancing API to develop web applications that use load balancing. For details, see the “About Web API Clients” chapter in the eServices Web API Client Developer’s Guide.

Interaction Server

Interaction Server can balance among multiple Universal Routing Servers and among eServices application servers. It can also connect to its database via multiple Database Access Points (DAPs). It does not use a user-accessible API for load balancing.

Balancing Universal Routing Servers

Interaction Server can balance among multiple instances of URS. This balancing proceeds by strategy: when an interaction reaches a strategy object in a workflow, Interaction Server selects (in round-robin fashion) from among all URS instances that have that strategy loaded.
To enable this type of load balancing, you must:

- Configure a connection from each URS to Interaction Server.
- For all participating URS instances, set the `agent_reservation` option to `true`.
- On the `Annex` tab of Interaction Server, set the `agent_reservation` option to `true` for the Application names of all participating URS instances.

> **Important**
> For additional information on the `agent_reservation` option, see the "Configuration Options" chapter in the Universal Routing 8.1 Reference Manual and "System Availability and Redundancy" in the Universal Routing Deployment Guide.

- Choose each URS when activating each strategy in Interaction Routing Designer (IRD). Do this by shift-clicking all of the desired URS instances in the `Choose Routing Server` window of the Strategy Activation Wizard.

Suppose Interaction Server has two URS instances connected to it: URS 1 has Strategies A and C loaded, and URS 2 has Strategies B and C loaded. Then,

- For interactions that arrive at Strategy A in a workflow, Interaction Server submits them to URS 1.
- For interactions that arrive at Strategy B in a workflow, Interaction Server submits them to URS 2.
- For interactions that arrive at Strategy C in a workflow, Interaction Server balances between URS 1 and URS 2.

If any instance of URS shuts down, Interaction Server detects that this instance is not available. If any interactions were pending in the unavailable URS, Interaction Server resubmits them to an available URS that has the required strategies loaded.

Balancing eServices Application Servers

An *application server* is a server that Interaction Server invokes when triggered to do so by a routing strategy. For example, a Classify object in a strategy triggers Interaction Server to invoke Classification Server. To do this, Interaction Server uses a protocol called *External Services Protocol* or ESP; therefore these servers are also called ESP servers.
The application servers that Interaction Server can balance among are:

- Classification Server.
- E-mail Server.
- SMS Server.
- Social Messaging Server.

> **Important**
>
> E-mail Server, Chat Server, SMS Server, and Social Messaging Server have a dual role, as follows:
>
> - When Interaction Server contacts these servers using ESP (for example, asking E-mail Server to generate an autoresponse), they are application servers (ESP servers) and Interaction Server is their client.
> - When these servers contact Interaction Server, using the Interaction Management Protocol, and ask to submit an incoming interaction, they are media servers and clients of Interaction Server.
>
> For more information on these protocols, see the Genesys Events and Models Reference Manual

Balancing Directly

You can load balance by configuring connections from Interaction Server directly to each instance of the application server.

This method encounters a limitation with multiple custom ESP servers that provide different service types. Custom ESP servers generally have the Third Party Server Application type in the Configuration Layer. This means that if, for example, you have several custom servers handling fax interactions and several custom servers handling IM interactions, and you configure them for load balancing by making direct connections, Interaction Server will be unable to distinguish the ones that handle fax from the ones that handle IM, and will therefore send fax requests to the IM servers and vice versa. The solution for this is to use Application Clusters, described in the next section.

Balancing Using Application Clusters

Starting in release 8.0.0, eServices supports the use of Application Clusters for ESP servers. You can configure a separate Application Cluster for each type of ESP server. Then a client such as

URS can call on the appropriate Application Cluster by name.

> **Important**
> Only one level of Application Clusters is allowed.

Balancing DB Servers

Interaction Server must work with only one database. However, Interaction Server supports multiple DAP connections to the same database through different DB Servers. You can configure this using multiple connections (DAPs) to one database.

> **Important**
> If for some reason the configuration of Interaction Server's DAP is erroneously changed to point to a different database, Interaction Server overlooks the error: it sends an error message saying that the configuration has changed, and continues to work with the original database. But if, in this same scenario, a switchover to a backup Interaction Server occurs, the backup Interaction Server has no way of knowing that the DAP configuration has (erroneously) changed, so it connects to the new database and starts sending requests to it. It must be emphasized that Genesys recommends that you do not change the DAP configuration on the fly.

Capture Points

Capture Points are integrated into Interaction Server in the 8.0.2 release. Therefore, the application host in the Capture Point configuration is not taken into account and the host of Interaction Server is used.

A Capture Point can be configured as a primary/backup pair. In this case, the host of the primary application must be the same as host of the primary Interaction Server and the host of the backup application must be the same as host of the backup Interaction Server.

The primary Interaction Server (by configuration) will search for the primary Capture Point application and use its configuration to start the capture point. The backup Interaction Server (by configuration) will search for the backup Capture Point application and use its configuration to start the Capture Point. If there is no backup Capture Point configured, the backup Interaction Server will use the primary Capture Point application.

Generally, there is no need to configure a backup Capture Point application in Configuration Manager or Genesys Administrator; a backup Interaction Server will start the "backup" Capture Point instances.

> **Important**
>
> The JMS Capture Point is the only integrated capture point supported in Interaction Server 8.0.2. The File Capture Point is new in Interaction Server 8.0.21, the Database Capture Point is added in release 8.1.0, and the Web Service Capture Point is added in release 8.1.2.

SMS Server

SMS Server is scalable if necessary in order to deal with high volumes of inbound SMS messages arriving from an SMS Center.
Genesys suggests the following method of scaling:

- Take the set of telephone numbers from which SMS messages can arrive and divide it into subsets. For example, you can define one subset as numbers in the (650) area code, and a second one as numbers in the (925) and (510) area codes.
- Deploy one SMS Server for each subset.
- Be sure that every served telephone number belongs to exactly one subset.
- Have all of the SMS Servers running simultaneously, to serve all inbound traffic.

Backup Configuration

This topic lists the types of backup configuration supported in eServices 8.1.

Component	Type of Support	Remarks
Chat Server	Warm standby and load balancing	Supported through load balancing on Web API Server and SMS Server.
Classification Server	Warm standby and load balancing	
Co-Browsing Server	Load balancing	Supported through load balancing on Web API Server and SMS Server.
E-mail Server	Warm standby and load balancing	Supported through load balancing on Web API Server and SMS Server; also through ESP load balancing by Interaction Server.
Interaction Server	Warm standby	See the Warning below.
Interaction Server Proxy	Warm standby	
SMS Server	Warm standby	
Social Messaging Server	Warm standby	
Training Server	Load balancing	Supported in that it can process multiple training jobs. However, if an instance of Training Server becomes unavailable while it is processing a job, then a second running instance of Training Server will not pick up the job for processing. Instead, you must restart the first instance.
Universal Contact Server	Warm standby	
Universal Contact Server Proxy	Warm standby	
Web API Server	Load balancing	Supported through load balancing on Web API Server and SMS Server.

Warning

For Interaction Server, Local Control Agent

- Must be running on the hosts of both primary and backup server.
- Must be connected to Solution Control Server.

For general information on warm standby, see the Framework Architecture Help and the Framework Deployment Guide.

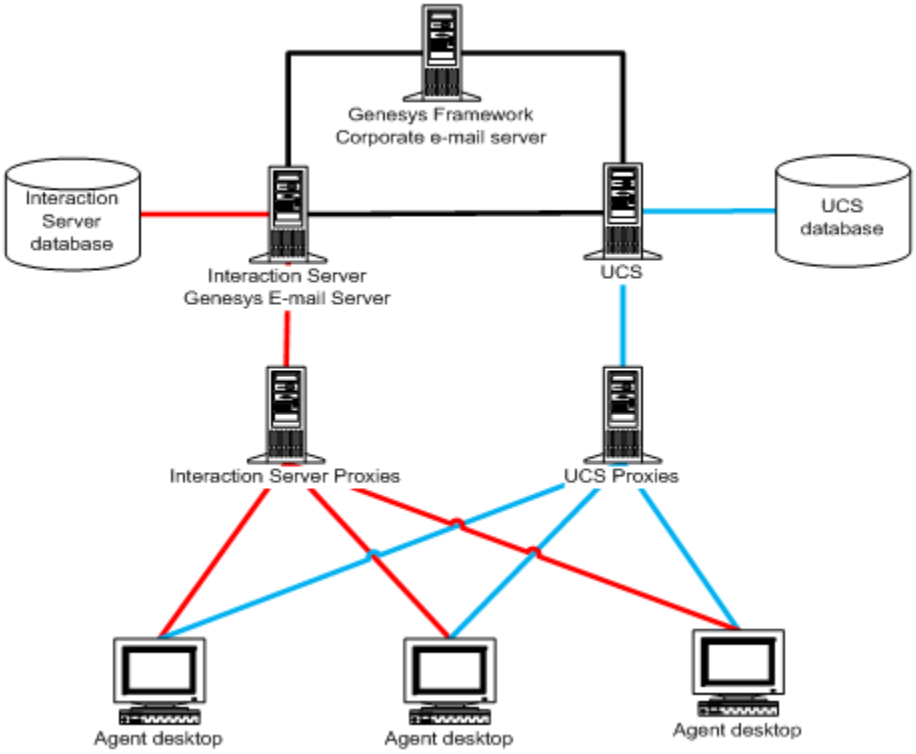
UCS and Interaction Server Proxies

Large numbers of custom desktop (ESP client) connections to Interaction Server and UCS may give rise to performance issues. To mitigate the issues caused by a high load on the server, Genesys introduced Interaction Server Proxy and UCS Proxy in release 7.6.1. Desktop applications can be configured to connect to these Proxies instead of the main server, significantly reducing the load on the server. For example, it is easier for the server to handle 20,000 clients that operate through ten proxies (only ten connections) than to handle the same 20,000 clients that each connect separately.

For a description of how to deploy these Proxy servers, see the "Manual Deployment-UCS Proxy, Interaction Server Proxy, and SMS Server" chapter of the eServices 8.1 Deployment Guide.

Because there are so many variables in deployment (choice of operating system, number of clients, details of architecture, and so on), it is not possible to provide exact guidelines as to when deploying a Proxy server would be advantageous. However it may be stated that you can anticipate performance issues when the number of clients exceeds 10,000.

The diagram below shows a sample deployment using both Interaction Server Proxy and UCS Proxy. Each of the agent desktops in the diagram can represent several thousand agents.



Sample Architecture Using Proxy Servers