**Components:**

**CS: CS master Priamry, CS maser backup, cs proxy**

**SCS : 0 or several(in distributed mode) Each server** controls it’s own subset of hosts, apps, different SC communicates with each other throught dedicated MS**.**

**SCI: as many as need. Has app in CME. Launch only one instance of SCI per host computer.**

**you can configure SCI to work with more than one Solution Control Server and more than one Log Database, SCI can only work with one SCS and one Log Database at a time.**

**LCA – has no backup**

**You cannot manually switch over applications of these types:**

**• Configuration Server**

**• Database Access Point**

**• Solution Control Server**

**HA**

**Configuration server(WS), Message server(WS), SCS(WS), SNMPMA(hot standby for genesis and not specified for net), CS proxy(WS), TServers(WS, Sc 7.1 - HS), StatServer, GAX(with LB)**

**IN SCS we are configuring HA of : CFGserver, MS**

**External autentification systems:** RADIUS and LDAP.

***New***

**in 8.5.1:**

**Alternate Disaster Recovery Deployment Model:** Management Framework now supports database replication leveraging the Microsoft SQL Server AlwaysOn

**Updated Language Pack support**

**Consolidation of Single-Tenant (Enterprise) and Multi-Tenant Configuration Server:** The single-tenant and multi-tenant Configuration Server IPs have been consolidated into one Configuration Server IP, single-tenant configurations can no longer be deployed for new installations

**Support of Load-Balancing by Configuration Server Proxy:** Clusters of Configuration Server Proxies can now support load balancing using the F5 load-balancer for agent-facing applications

**More secure and efficient agent data retrieval:** Configuration Server can now be configured to execute requests, submitted via an internal brief API, taking into account permissions of the parties involved.

* **New component:** The Configuration Database Maintenance Scripts component enables upgrade of the Configuration Database schema and locale as needed. Additional scripts are included in this component to help you maintain and analyze the Configuration Database.

**New and modified scripts for initalizing Log Database:** A new Message Server script, **drop\_tables\_<DBMS>.sql** drops existing tables and procedures.

* **Genesys Deployment Agent not installed automatically:** The Genesys Deployment Agent is no longer automatically installed with LCA by default.
* **New SNMP implementation:** You can now use Net-SNMP, a third-party tool, to implement SNMP in Genesys software. Net-SNMP can be used in parallel with, or instead of, Genesys SNMP Master Agent, and provides the same functionality.

**New mlcmd parameter:** The new **mlcmd** parameter -start-initapp starts an application without waiting for the status to change to Initializing/Running

**Configuration Server port segmentation:** You can configure separate ports on Configuration Server and Configuration Server Proxy that are restricted for use only by client User Interface (UI) type applications

**Secure Connections for Centralized Log:** Configuration Server and Solution Control Server can now use Simple or Mutual TLS to secure a connection to the Message Server designated for the Centralized Log.

* The password for the Configuration Database can now be encoded using an asymmetric encryption algorithm, with the encryption and decryption keys stored in external files.
* **Enforcement of External Authentication at Tenant Level:** External authentication can now be selectively enforced at the Tenant level in a multi–tenant environment.

Starting in 8.5.1, you can use an environment variable with Configuration Server, Solution Control Server, and Message Server to indicate how each component is to handle network communications, using either single-threading or multi-threading. The three system environment variables are:

• Configuration Server—**GCTI\_CONFSERV\_CONN\_STARTUP\_DEFAULT**

• Solution Control Server—**GCTI\_SCS\_CONN\_STARTUP\_DEFAULT**

• Message Server Server—**GCTI\_MS\_CONN\_STARTUP\_DEFAULT**

[New in Release 8.5.0](javascript:toggleDisplay(%20%22toggledisplay2%22,%20%22[-]%20New%20in%20Release%208.5.0%22,%20%22[+]%20New%20in%20Release%208.5.0%22%20))

* DB Server is no longer required for Configuration Server to access the Configuration Database.
* Configuration Server now requires a valid license file before starting for the first time.
* Support for Genesys License Reporting Manager (LRM).
* DB Server is no longer required for Message Server to access the Log Database.

Starting in release 8.0, each Solution Control Server in a Distributed Solution Control Serverenvironment can detect the failure of a remote site controlled by another Solution Control Server.

Starting in release 8.0, you can configure Configuration Server so that some Genesys GUI applications display the date and time of the previous login for the currently logged-in user.

Now, RADIUS and LDAP external authentication, starting in

release 8.0 and 8.1 respectively, can be configured on the Master Configuration Server and on each Configuration Server Proxy. Therefore, each Configuration Server Proxy can process authentication requests itself, and does not need to pass them on to the Master Configuration Server.

Starting in release 8.0, you can stop an Application or a Solution gracefully, known as a

*graceful shutdown* or *graceful stop*. Applications refuse any new requests, but continue to process their current requests. A Solution gracefully shuts down all of its composite

Applications, then stops.

Starting with release 8.0, application templates for some Genesys components come with additional XML files called *Application Metadata* files

Prior to release 8.5.1, Genesys used the RSA BSAFE SSL-C implementation of secure

protocols. Starting with release 8.5.1, the default implementation is replaced with the

OpenSSL library. Starting in release 8.5.1, Security Pack uses SHA1 as the default algorithm for signing certificates, with an option to use SHA256.

After database initialization

**Default user**

**• Four Application Template objects, as follows:**

◦ Configuration Server

◦ Configuration Manager

◦ Genesys Administrator

◦ Genesys Administrator Server

**• Five Application objects, as follows:**

◦ confserv object of type Configuration Server.

◦ default object of type Configuration Manager.

◦ Genesys Administrator object of type Genesys Administrator.

◦ Genesys AdministratorServer object of type Genesys Administrator

Server.

◦ Installation Configuration Utility Application object with the name set to

ITCUtility. This utility supports configuration updates during installation

processes for Genesys components. No additional configuration is needed..

• The default Access Group objects: Users, Administrators, and Super Administrators.

For more information, refer to Security Considerations.

• Folders for all types of objects managed by the Configuration Layer.

The Configuration Database also contains a number of other predefined objects (for

example, Alarm Conditions)

**Configuration Environment Types**

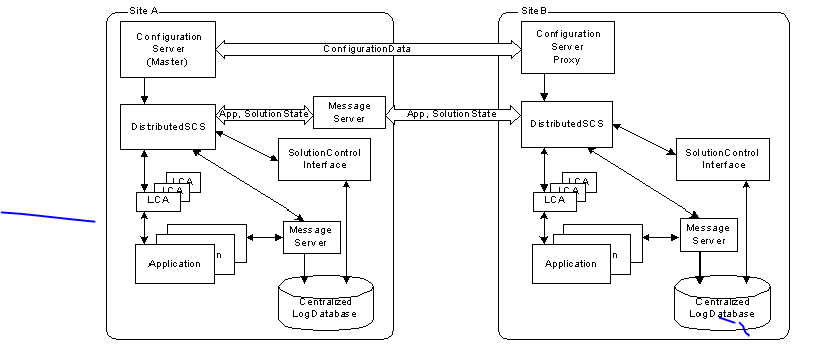
Genesys provides its software to two types of companies:

• Companies that own their telephony equipment and use it for their own needs.

• Companies (such as service providers) that make their telephony equipment available to other companies.

**One switch – one T-Server**

**Configuration Server Proxy**



Using Configuration Server Proxy increases the robustness of the whole system, decreases the number of client connections to Configuration Server, and minimizes network traffic.

When Configuration Server Proxy is configured, existing clients can continue, and new

clients start, their operations when Configuration Server fails.

**How it Works**

Configuration Server Proxies at multiple remote sites are connecting to the master Configuration Server. Configuration Server Proxy passes messages to and from

Configuration Server. Moreover, the proxy keeps the configuration data in its memory and responds to client data requests. Any configuration data updates are passed immediately to Configuration Server Proxy, so that it is always up to date; no additional configuration is required to specify an update interval.

**Configuration Server Proxy Functions**

Receives subscription requests from clients and handles them without passing the

requests to Configuration Server

• Stores in internal memory all configuration data it receives from Configuration

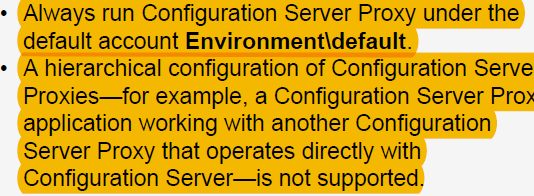
Server.

• Receives notifications on data changes from Configuration Server, updates internal

memory, and passes notifications to clients.

• Receives read-data requests from clients and responds to them using the data

stored in the internal memory.



If you want this Configuration Server Proxy to be writable, set the option

**proxy-writable** in the **[csproxy]** section to true.

HA Configuration Server Proxy supports ADDP between the pair of proxy servers if ADDP has been enabled between the master Configuration Server and Configuration Server Proxy in the Connections tab of the proxy server.

**Writable Configuration Server Proxies**

By default, Configuration Server Proxy provides read-only access to configuration data.

Configuration Server clients that require write access to Configuration Server must still

connect directly to Configuration Server.

If the connection between the main and proxy servers is lost, and ADDP is configured

between Configuration Server Proxy and the main Configuration Server and also between the proxy server and its client, the session is not restored. Clients of the Configuration Server Proxy must reregister and read all data from scratch.

**Config proxy LB clusters**

****

Starting and Stopping Framework Components

You can start and stop a Framework component in any of the following ways:

• Use the startup file(**bat**) created by the installation script. This file can only be used to

start the component, you must use one of the other ways to stop the component.

• Use the Management Layer.

• Manually, specify command-line parameters.

• Use the Windows Services Manager, available only in Windows.

**TENANTS**

**The enterprise** (also referred to as single-tenant) configuration environment serves the needs of a single company that owns its telephony equipment and uses it for its own needs.

**The hierarchical multi-tenant** configuration environment serves the needs of a company—typically, a service provider—making its telephony equipment available to other companies. So, this configuration environment also serves the needs of every company using the service. In this environment, configuration information about the resources that are managed exclusively by the service provider is visible on the service provider side only. Only personnel from the service provider company can register the entities that provide the technical foundation for setting up the CTI services, such as switching offices, data network hosts, and CTI applications

Genesys defines a large configuration environment as one in which the Configuration Database stores 50,000 or more configuration objects.

Consider using Configuration Unit and Folder objects when creating a large number of configuration objects. The recommended number of configuration objects per folder is up to 4,000. Anything larger significantly increases Configuration Manager time for loading configuration objects.

When creating configuration objects of the Script type (for example, routing strategies), keep in mind that both the number of Script objects and the script size significantly affect the time it takes Configuration Manager to load the Script configuration objects. If you create large scripts, reduce the number of Script objects in a subfolder to achieve an acceptable loading speed. For instance, for the script-type configuration objects approximately 150 KB in size, limiting the number of script-type objects to 30 per subfolder guarantees an acceptable loading speed.

**Multi-language Environments (UTF-8 Support)**

In release **8.1.2,** Configuration Server introduced this functionality for string fields of some configuration objects (Business Attribute and Attribute Value objects). In release **8.1.3**, this support was extended to all string fields of all configuration objects, with some exceptions noted later in this section. This functionality is optional, and must be enabled to take effect.

You must enable Configuration Server to handle UTF-8 data within the following time frame: • After the database has been initialized. • Before you start Configuration Server for the first time. Trying to enable a Configuration Server for UTF-8 that has already been started at least once will result in a warning message being logged.

Framework **does not support** UTF-8 encoding of the following items; they must be in ASCII.

• Names of Application objects

• Command-line arguments specified during configuration in the Start Info section

• Command-line arguments used by mlcmd and logutility

• Local configuration file used by Configuration Server

• SNMP traps and scalar data

• Database access credentials in the [dbserver] configuration option section of DB Server

• Database parameters in Database Access Points

• Host names

• Database table name in Table Access objects

• Log names and log file names specified in the [log] configuration option section

**Planning**

**Configuration Layer**

**• Configuration Server treats its information and checks integrity constraints in a case-sensitive manner. the most logical location for it is on the computer running DBServer.**

**Management layer**

**• Solution and application control and monitoring**

Local Control Agent

• Solution Control Server

• Solution Control Interface (optional)

**• Centralized logging**

• Centralized Log Database

• DBServer (as a client of Configuration Server)

• Message Server

• Solution Control Interface (optional)

**• Alarm signaling**

• Message Server

• Solution Control Server

• Solution Control Interface (optional)

• Genesys SNMP Master Agent, if SNMP alarm signaling is required. See also “Built-in SNMP Support” on page52.

**• Application failure management**

Local Control Agent

• Solution Control Server

• Solution Control Interface (optional)

**Although any number of Message Servers can store log records in the same Log Database, one Message Server cannot store log records to more than one Log Database.**

**If you want Message Server to provide alarms, you must connect it to Solution Control Server. This means that you must configure a connection to every Message Server in the SCS Application object’s Properties dialog box.**

**Service layer**:

Consider dividing database-related traffic evenly among any number of DBServers, each serving up to 255 clients.

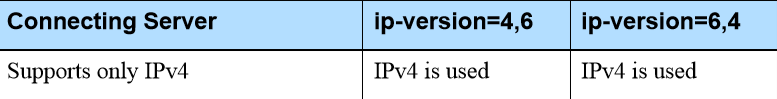
**IPv4, IPv6**

1)Environment variable

2) enable-ipv6 option in the common section of the options of the component’s Application object.

To configure this choice(which version will be used), use the Transport parameter ip-version on the Advanced tab of the Connection Info dialog box for the connection:

ip-version Default Value: 4,6 Valid Values: 4,6 and 6,4



**Configuration History Log**

The Configuration History Log consists of a set of a records that contains historical information about client sessions and changes to configuration objects.

For all Configuration Servers except Configuration Server Proxies, the records are stored in the Configuration Database. For Configuration Server Proxy, the records are stored in a separate database referred to as the History Log Database. The History Log is installed with default parameters when you install Configuration Server or Configuration Server Proxy. You configure the

History Log parameters in the options of the Configuration Server Application object in Genesys Administrator or Configuration Manager.

**Database Failures**

If a response is not received within the interval specified by the configuration option db-request-timeout, the client process stops executing. This is understood by DB Server as a failure of the DBMS, and it tries to reconnect.

The option db-request-timeout is configured in the DB Server Application object via the Query Timeout field for Database Access Point (DAP)

User Authorization

Starting in release 8.0, an additional layer of security is available through Genesys Administrator, called Role-Based Access Control. This enables the

Note: The inactivity feature survives reconnection timeouts. In other words, if the interface application becomes disconnected from Configuration Server after the forced re-login timeout has expired but before the user has logged in again, the user must still log in before he or she can access the system.

system administrator (or a designated individual) to define access to objects based on what is to be done (viewed, modified, deleted) to the objects.

**Each port can have one of the following listening modes:** • unsecured—The port is not secured by TLS. This is the default status of a port. • secured—The port is secured by TLS. • auto-detect—This status applies only to ports on the Configuration Server, and is used only when configuring secure connections to the Configuration Server. If an application that is trying to connect to an auto-detect port has security settings specified in its configuration, Configuration Server checks the validity of those settings. Depending on the results, the client will be connected in secure or unsecured mode.

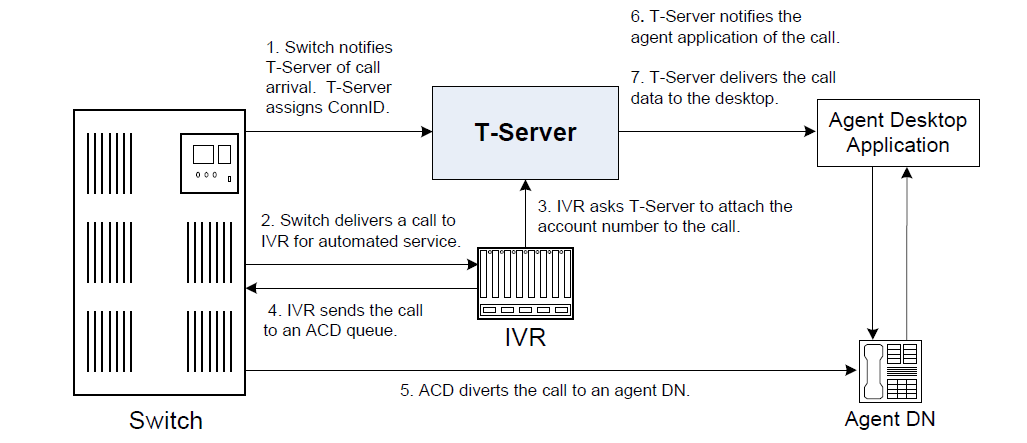
**T-servers**

T-Server has several typical clients: Stat Server, Interaction Concentrator,

Universal Routing Server, and agent desktop applications. T-Server is a TCP/IP server that enables intelligent communication between media-specific protocols (such as the various CTI protocols, including CSTA and ASAI) and TCP/IP-based clients of T-Server.

On the client-application end, T-Server offers three models (call model, agent model, and device model) unified for all switches. The core functionality (suchas processing an inbound call, an agent login, or a call-forwarding request) translates into a unified application programming interface (API) called T-Library, so that applications do not need to know what specific switch model

they are dealing with.



**Step 1**

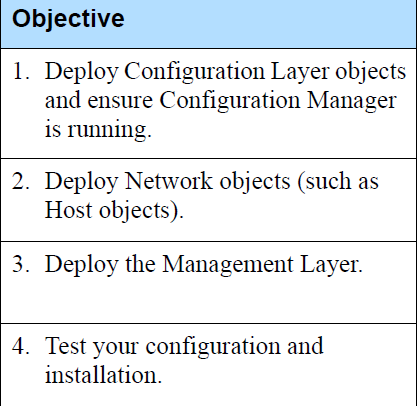
When the call arrives at the switch, T-Server creates a call in its internal structure. T-Server assigns the call a unique identifier, connection ID.

Since the 6.0 release of T-Server, the Advanced Disconnect Detection Protocol (ADDP) has replaced the Keep-Alive Protocol (KPL) as the method to detect

**Software Requirements**

Configuration Layer, If you intend to monitor or control T-Server through the Management Layer, you must also install and configure components of this Framework layer, such as Local Control Agent (LCA), Message Server, Solution Control Server (SCS), and Solution Control

Interface (SCI), before deploying T-Server.



Deploying T-Server manually requires that you configure a number of different objects in the Configuration Layer prior to setting up your T-Server objects and then install T-Server:

**Switching Offices, Switches, DNs and Agent Logins**

**Genesys Events**

**Participants**

*Participants* are software components that send and receive messages. The

participants in the models in this book include the following:

**•** T-Server and IVR Server

**•** Switch and IVR

**•** Interaction Server

**•** Media server (E-mail Server Java, Chat Server, or a custom application)

**•** Agent application

**•** Universal Routing Server

**•** Reporting engine (Stat Server or Call Concentrator)

**GAX**

GAX only supports the 64-bit version of Oracle Java HotSpot Server VM.

GAX must have a connection to Solution Control Server (SCS) for the System Dashboard to function.

**GA**

**Pulse**

On the Pulse dashboard, widgets display user-defined Donut, Grid, Key Performance

Indicator (KPI), or List charts of statistics for objects

****

**Pulse Collector**

Pulse Collector is a background near-realtime statistical data collection and processing

engine. It performs the following activities:

• Reads the metadata from the Pulse database upon startup and whenever changes

are made to report definitions in Pulse.

• Uses the report definitions stored in the Pulse database to determine which statistics

and objects to include.

• Creates snapshots with current data from Stat Server and formula-based statistics

calculated by Pulse Collector, on the specified file system for reference by Pulse.

• Maintains a constant connection with Configuration Server to retrieve changes,

additions, and deletions to configuration objects.

**Pulse**

Pulse is a GAX plug-in that performs the following activities:

• Handles user authentication and permissions validation.

• Filters and delivers report data according to the permissions and tenancy of the user

who is requesting the data.

Displays report content in widgets, such as the listing and content of reports.

• Saves the report definitions to the Pulse Database, which it shares with Pulse

Collector.

**Pulse interacts with several products within the Genesys Framework to provide real-time snapshots of contact center data:**

**Configuration** **Server** provides the following data to Pulse Collector:

• Information about the existence of contact center objects (for example, tenants,

agents, places, calling lists, or campaigns)

• Statistical parameters (for example, time ranges, time profiles, filters, and statistical

types)

• Information about changes to contact center objects (for example, a deleted agent, a

renamed queue, or a new Agent Group).

**Pulse Collector uses this data to provide content for Pulse.**

**Stat Server**

As a client of Stat Server, Pulse Collector requests statistics for objects belonging to

particular reports. Stat Server supplies both information about the interactions that the

objects that handle as well as noninteraction-related data about the objects themselves (for example, agent status). Pulse Collector returns information for all stattypes that are configured in the options of all Stat Servers to which Collector is connected.

**DB Server**

As a client of DB Server, Pulse Collector reads information about active widgets and updates the layout statuses, when layout status changes occur within the Pulse Collector.