white paper

Deploying SCC as a High Availability Solution

A guide for implementers of large-scale systems using Softdial Contact Center™

sytel

white paper

Contents:

1. Introduction 1

1.1 Technology vs Process 1

1.2 Outage and Outage Cost 2

1.3 The SCC Approach to Redundancy 3

1.3.1 Process Separation 3

1.3.2 Multi-Instancing 3

1.3.3 Clustering 4

1.3.4 Database session reliability 4

1.3.5 Campaign/Agent/Telephony session reliability 4

2. A Prescriptive Approach to High Availability 5

2.1 Mapping Server Roles to Hosts 6

2.1.1 Small Installations 6

2.1.2 Low-Volume MSP or Large Installation 7

2.1.3 High-Volume MSP 7

2.2 High-Availability Strategy 7

2.3 Redundancy 9

2.4 Configuring Replication 9

2.4.1 Replication of Service Configuration 10

2.4.2 Configuring `srep` 11

2.5 Configuring Failover 12

2.5.1. Failing Over onto a Backup Host 12

2.5.2. Configuring the Backup Host 12

2.5.3. Configuring `smon` 12

2.5.4. Operating Procedures 13

2.5.5. Failing Back onto a Primary Host 13

2.6 Telephony Clustering 14

2.6.1. Configuring STG Clustering 16

2.7 Database scaling & resilience 17

2.7.1. Scalability via Sharding 18

2.7.2. Redundancy via Replica Sets 18

2.8 Web Application Scalability and Resilience 20

2.8.1 Session persistence 21

2.8.2 Redundancy 21

2.9 HTTP tunnelling using NetClient and NetBridge....22
2.8.1 Setting up NetBridge....22
2.8.2 Setting up NetClient....22

3. Hardware and Network Resilience 23

3.1 Servers....24
3.1.1. Virtual Servers....24
3.1.2. Physical Servers....24
3.1.3. Server Sourcing....24

3.2 Managing Resources....25
3.2.1. No other applications deployed on SCC hosts....26
3.2.2. Separate Log Volume....26
3.2.3. Log purge....26
3.2.4. Scheduled Service Restarts....26

3.3 Network Infrastructure....27

3.4 Power....27

3.5 Bearer Networks....28
3.5.1. Multiple SIP Service Providers....28
3.5.2. Multiple TDM Service Providers....29

4. Proving your High-Availability Solution 29

Appendix A – `Smon` Configuration Options 31

Appendix B – Sample Failback Batch Script 32

Appendix C – Cluster Configuration Options 33

Appendix D – Tunneling Configuration Options 34

white paper

1. Introduction

Softdial Contact Center (SCC) is deployed to provide contact center telephony and applications infrastructure for managed service providers and contact center services for large contact centers.

Successful deployment on a large scale requires three things:

1) Technology that enables zero or minimum cost failover.
2) A deployment architecture that avoids limit conditions.
3) Applications designed and implemented in such a way as to minimise the cost of errors.

The third point is included for the sake of completeness. This is not the subject of this paper.

This document explains how SCC provides a technology solution for resilience and how best to deploy SCC in order to achieve high system availability.

It is assumed that the reader has a basic understanding of how SCC is implemented as a suite of network services that work co-operatively to deliver a contact center solution.

This document also assumes that all components of SCC will be deployed. Sytel Partners who deploy their own Telephony and application components against SCC may use this document as a guide but

1.1 Technology vs Process

As a manufacturer we are often asked the question 'Is your system resilient/redundant/high availability?'

This question reflects the valid concerns that a customer has about a technology supplier's ability to deliver high availability. It is fair to say that SCC provides all the necessary infrastructure and tools to support high availability. However deploying a HA solution involves investment in hardware infrastructure , additional configuration and operational proving of the configured system.

A high availability solution has to be built on top of a redundant software architecture by implementing infrastructure that enables continuous service to be delivered in the event of process, server, network and bearer failures. These things are all beyond Sytel's control.

The good news is that Sytel has much experience in helping large customers configure a high availability solution based on our technology underpinnings, which make the job of providing HA somewhat easier.

The document prescribes infrastructure and operating process that customers should adopt in order to ensure system availability can be as good as the technology can provide. Throughout this document infrastructure and process recommendations will be highlighted in italics.

1.2 Outage and Outage Cost

If a customer does everything right in terms of implementing...

- redundancy at all levels within SCC
- a no single point of failure hardware architecture
- connections to multiple bearer networks
- appropriate levels of system monitoring
- strict change control

...then ultra high availability can be achieved, measuring outage per year in minutes and seconds only. Note that redundancy in SCC is only one of 5 things that need to be implemented.

Full redundancy in SCC means that in the worst-case event of a catastrophic server or service failure, service will be restored automatically in less than one minute.

Not implementing redundancy (but having failover mechanism involving manual intervention) reduces availability by one order of magnitude.

Not implementing redundancy and not having a failover plan reduces availability by two orders of magnitude.

Section 2 below – A Prescriptive Approach to High Availability - describes the redundancy model Sytel recommends to its partners and customers.

Section 3 below – Hardware and Network Resilience - provides recommendations on how to implement a resilient hardware and network architecture.

white paper

1.3 The SCC Approach to Redundancy

A redundant architecture will not protect you from the cost and reputation damage of outage if failure of a component within your platform results in outage for a substantial part of your user community or for a substantial amount of time. Sytel's redundancy solution is built to solve this problem.

The SCC approach to solving this problem is threefold:

1) Process separation. The server-side components of SCC are divided into discrete services each delivering a specific function. This minimises complexity of components and therefore failure rates and failure cost.

2) Multiplexing; running one activity across several physical processes. There are two types of multiplexing employed within SCC; Multi-instancing - used primarily for tenant segregation and Clustering - Used for telephony.

3) Session reliability. State of resources such as database services, campaign and agent endpoints are reliant on reliable communications. Sytel provides various tools to support session reliability.

1.3.1 Process Separation

Processes are broken down by business function to minimise complexity and also to ring-fence complex and failure-prone operations.

A good example of this is campaign manager. It functions as a database proxy and has code to manage database transaction failure. It also publishes interfaces that other SCC services use to take advantage of its capabilities. A script running in scripter will drive database activities through campaign manager rather than accessing the database from scripter directly.

This means that there is only one application that has to implement complex code for managing database transaction failure but its capabilities can be used by other applications.

This principle is applied across the SCC services at large.

1.3.2 Multi-Instancing

Application services such as Campaign Manager, Scripter, Publisher, Scheduler run an instance per tenant, as well as in some cases a landlord instance.

By multi-instancing the load on each service is reduced, and likelihood and cost of failure is reduced.

1.3.3 Clustering

Clustering is used for telephony resources. The SCC telephony model is B2BUA. This means that each agent and each customer involved in a telephony transaction has a separate connection to a specific telephony server.

Because of the switching model employed it is sometimes necessary to bridge calls between instances of telephony servers to connect a customer to an agent. CallGem makes decisions on bridging by executing a script that has certain default behaviours, but can be customised to specific end-user topologies.

Deploying a redundant cluster typically involves running a standard script for making bridging decisions and deploying N+1 telephony servers to deliver N telephony servers worth of capacity.

Clustering is discussed in more detail in section 2.6.

1.3.4 Database session reliability

Database session reliability measures are built in to Softdial Campaign Manager and Softdial Publisher, the two services that act as database proxies within the SCC architecture. These provide for rollforward recovery in the event of database connection failure. Rollforward recovery is available in SCC Version 10.5 and later versions of SCC.

1.3.5 Campaign/Agent/Telephony session reliability

Third-party services that deliver campaign, agent and telephony sessions, as well as Sytel's own services (CM, Scripter, STG) can mitigate the possibility of resource outage caused by connection failure by employing a HTTP tunnel. The HA kit contains Softdial NetBridge and NetClient, a pair of services that deliver HTTP tunnelling for services that communicate using the SDMP protocol. Setup for HTTP tunnelling is discussed in section 2.9.

2. A Prescriptive Approach to High Availability

In order to deliver resilient services on a large scale it is necessary to prescribe system configuration. Sytel's prescriptive architecture is based on experience of deploying SCC in volume installations.

The basic premise is that SCC components may fulfil one of 6 roles.

The roles are:

1) Controller server
2) Web server
3) Application server
4) Database server
5) Telephony server
6) Dictionary server

NOTE: A server role is a group of linked tasks or services. It does not necessarily mean a separate physical or virtual host is required.

Each of these roles corresponds to the running of a number of SCC services. In a trivial installation all services can be on the same host. This will work for demonstration and development purposes but is not recommended for live operation.

The breakdown of servers into roles enables us to build a resilience plan based around the function that the server undertakes. Some roles (Controller and Telephony) are real-time. By making this separation it enables us to craft resilience measures that work without generating a huge server and admin footprint.

Relationship between server roles and SCC Services:

Role	Services
Controller	CallGem Namespace NetBridge (****) Scheduler (landlord) Workflow Server (*)
Web Server	SoftdialWebServer (Apache) SoftdialMySQL (MySQL instance to support web server) HTTPSdmpBridge RecordMonitor

Application Server	CM Server (instance per tenant) Scripter Server (per tenant) Scheduler (per tenant) Publisher (per tenant) Dictionary Service (per tenant) NetClient (****)
Database Server	Publisher (landlord) MongoDB Relational database of customer's choice (**)
Telephony Server	STG 3rd Party Speech provider
Dictionary server (***)	Landlord dictionary service instances

Notes:
(*) Workflow Server will in the near future be a tenant-side service with tenant service configurations.
(**) The database server may be used to host the application data store or it may be delivered by a customer's existing server estate.
(***) The dictionary server is strictly optional.
(****) NetBridge and NetClient form a HTTP tunnel to deliver reliable sessions over unreliable network transports.

2.1 Mapping Server Roles to Hosts

It should also be noted that in smaller installations some of these roles may be combined in one physical server or VM instance. In the absence of any overriding considerations we would recommend the following mapping:

2.1.1 Small Installations

In a small production installation the minimum number of servers / VM instances is 3:

Server 1: Controller, Web Server and Application Server roles
Server 2: Database Server
Server 3: Telephony server

Given an 8-core, 32GB memory host this server configuration will support up to 200 agents. This model is typically used for CPE installations without redundancy.

2.1.2 Low-Volume MSP or Large Installation

A more typical deployment separates the Controller and application server roles

Server 1: Controller and Web Server roles
Server 2: Application Server
Server 3: Database Server
Server 4: Telephony server

In this case there may be multiple database and telephony server instances but only one instance of Servers 1 and 2. Redundancy should generally be implemented for any installation above 200 users in size.

This model is suited to low volume managed service provision (to 600 agents / 2000 channels). Redundancy can be configured at low cost.

2.1.3 High-Volume MSP

For large deployments (upwards of 2000 channels) the controller and web server roles need to be split onto separate hosts. This prevents the Controller from becoming an IO bottleneck and provides possibilities for network load balancing for web application payload. This gives us:

Server 1: Controller
Server 2: Web Server
Server 3: Application Server
Server 4: Database Server
Server 5: Telephony server

This model involves multiple instances of application, Database and Telephony servers and MUST (RFC2119) implement redundancy across all server roles as discussed in the next section.

2.2 High-Availability Strategy

To achieve high availability, redundancy and automated failover of services is necessary. This is complicated by the fact that the Controller and Communications servers run real-time, stateful applications that are not easily replicated.

Each server role has specific redundancy and failover options, described in the table below:

Role	Strategy

Controller	1+1 redundancy of all services. Auto failover to backup server on host failure. Auto restart on service failure.
Web Server	Apache/MySQL/HTTPSdmpBridge can be multi-instanced and load-balanced. RecordMonitor is 1+1 redundant. In a HA solution at least 2 web servers will need to be employed.
Application Server	1+1 redundancy of all services. Auto failover to backup server on host failure. Auto restart on service failure. In a large installation multiple application servers are likely. It is possible to configure multiple application servers to have the same backup server.
Database Server	1+1 redundancy on Publisher (landlord) MongoDB replication Database replication on customer RDBMS.
Telephony Server	STG is multi-instanced and is load-balanced through the controller. This enables N+1 redundant deployment of telephony servers.
Dictionary server (***)	1+1 redundancy on landlord dictionaries Auto failover to backup server on host failure. Auto restart on service failure.

Sections 2.3-2.5 discuss configuration of redundancy of services.

Telephony needs be treated separately for high availability owing to the specific considerations of real-time streaming and interaction between telephony servers. Section 2.6 discusses configuration of telephony clustering.

2.3 Redundancy

SCC services have a standard pattern for redundancy.

In order to configure redundancy, static configurations for services need to replicated, and remote failover configured.

Normally in the event of a SCC service failure the service will fail over gracefully. This means that the service will auto-restart, reconnect with its bus controller and re-establish any resources it was responsible for.

Reasons for a SCC service failure will usually be environment-related. Running out of memory, disk and other physical resources can be managed. Best-practice resource management is discussed in §3.

In the event of a server failure, services will need to restart on another host. Since the service connects into a bus controller which provides access to the outside world, this changeover can be made automatic and seamless.

The standard hardware model for redundancy involves running multiple virtualisation hosts. Each host will run a number of primary server roles and some backup server roles. In the event of a physical host failure the overall system will fail over quickly and automatically so that service can continue to be delivered.

2.4 Configuring Replication

SCC runs as a series of Windows services that run on any x86 or x64-based Windows platform from Windows XP & Server 2003 through to Windows 7 and Server 2008 R2.

SCC services that have local configuration have this contained within an XML document. Those services that treat this configuration as dynamic do so by opening, reading and closing the configuration file on a timed basis.

This form of config access allows a customer to use replication technology of their choice to perform replication. Some customers may wish to use Windows Distributed File System replication which will work but will subject you to Operating System and licensing constraints that may not be desirable or practical.

Because file replication technology can be difficult to set up and maintain Sytel has produced its own lightweight file replication service, **`srep`**. `srep` interrogates the service configuration on a host and backs up configuration to a remote host using standard file system techniques.

srep is available as part of Sytel's HA deployment kit, available from Sytel support (support@sytelco.com)

Customers approaching file replication for the first time may wish to use Sytel's packaged or third-party open-source technology rather than using Windows DFS replication. DFS replication provides a robust solution for 2-way directory replication based on Windows Server and Active Directory, but is costly to set up and manage.

2.4.1 Replication of Service Configuration

This information is provided for those customers who have their own replication strategy. Sytel's packaged replication tool, `srep`, incorporates this information in its implementation and automatically replicates configuration for installed services.

Note `<RROOT>` is the root registry path for SCC and is:

For 32-bit hosts	`HKEY_LOCAL_MACHINE\Software\Sytel`
For 64-bit hosts	`HKEY_LOCAL_MACHINE\Software\Wow6432Node\Sytel`

This is for historical reasons and is constant whether using native 64-bit or 32-bit variants of software.

SCC Service	Static Configuration	Config File Replication
Campaign Manager (<tenant>)	Service registration Registry <RROOT>\CM\<tenant>	%SOFTDIAL_ROOT%\CM\<tenant> (and subfolders) replicated to same path on remote host.
Scripter 2 Server (<tenant>)	Service registration Registry <RROOT>\SCR2\<tenant>	%SOFTDIAL_ROOT%\Scripter2\Engine\ <tenant> (and subfolders) replicated to same path on remote host.
Publisher (<tenant>)	Service registration Registry <RROOT>\CM\<tenant>	%SOFTDIAL_ROOT%\PUB\ <tenant>\xml (and subfolders) replicated to same path on remote host.
Scheduler (<tenant>)	Service registration Registry <RROOT>\SCH\<tenant>	%SOFTDIAL_ROOT%\SCH\ Config.xml replicated to same path on remote host.
HTTPSDMPBridge	-	%SOFTDIAL_ROOT%\HSBridge\HttpSMDPBridgeService.exe.config
(TO Finish)		

2.4.2 Configuring `srep`

`srep` is a windows service that follows Sytel's standard install pattern. It should be installed on the primary host running a particular server role, and will run on this host, replicating configuration to a backup host.

`srep` Installation

To install `srep`, copy the `srep` folder from the deployment kit to `%SOFTDIAL_ROOT%` on the primary host.

Run a command prompt with administrator privileges and execute the following commands:

```
> CD %SOFTDIAL_ROOT%\srep
> srep install
```

This will register `srep` as an auto-start service running under the local machine's `SYSTEM` account.

Since `srep` will be using the local network file system to copy files from one host to another, it needs to run with the right privileges to enable access to the file system on the remote host. `srep` needs to run as a user (most likely a Windows domain user) with access rights to the file system on the remote host.

To change the use account that the service runs under, run the services.msc console snap-in. Right mouse click on the `srep` service entry and select properties. Select the 'Log On' tab and enter the user credentials

Note: You should consider setting up a user account specifically for this task with a password that does not expire.

`srep` Configuration

The srep configuration document (Config.xml) resides in the `%SOFTDIAL_ROOT%\srep` folder. It contains two configuration entries which identify the DNS host name/ip address of the backup host and a UNC path to the `%SOFTDIAL_ROOT%` folder on the remote machine.

A sample configuration document is shown below with the 2 configuration entries highlighted in red:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="sytelco.com/srepConfig">
```

```
 <remoteHostName>minime</remoteHostName>
 <remoteHostSoftdialRoot>\\minime\c$\Softdial</remoteHostSoftdialRoot>
</configuration>
```

In the example above the c$ administrative share is used. You may have to explicitly share and set permissions on the backup server `%SOFTDIAL_ROOT%` folder so that the account `srep` runs under can access the remote file system.

2.5 Configuring Failover

2.5.1. Failing Over onto a Backup Host

To make failover automatic Sytel provides a service that uses ICMP datagrams (same as used for ping) to check that a remote host is alive, and if not alive execute a series of commands to start services on the local host.

This enables failover to be automated in that each backup host monitors its primary and can then automatically start backup services using last known good configuration. This will also result in any failed resources being re-established.

The service is called **smon** and is packaged separately from the core SCC software. This is deliberate as configuration of server failover requires some manual configuration.

`smon` is available as part of Sytel's HA deployment kit.

Backup servers can be multi-purpose; in other words a backup server can act as backup for more than one other server. `smon` can be configured to monitor multiple hosts to limit redundancy costs. A word of caution though – in the event of server failover it means that not only is the failed over server no longer redundant, but also all other servers that the backup server was acting on behalf of are no longer redundant.

2.5.2. Configuring the Backup Host

The backup host will need to have all services installed that are installed on any of the primary servers for which it is acting as backup. In practice if using one backup server to provide failover for several primary servers it is a good idea to install all components of SCC on the machine

2.5.3. Configuring smon

`smon` is also a windows service that follows Sytel's standard install pattern. It

is installed on a server assigned as a backup for a primary server. It is possible to have a backup server act as backup for several different primary servers. The only server role that has to have dedicated server backup is the Controller, as on failover the backup machine will change its IP address.

smon Installation

To install `smon`, copy the `smon` folder from the deployment kit to `%SOFTDIAL_ROOT%` on the primary host.

Run a command prompt with administrator privileges and execute the following commands:

```
> CD %SOFTDIAL_ROOT%\smon
> smon install
```

This will register `smon` as an auto-start service running under the local machine's SYSTEM account.

smon Configuration

The configuration document for `smon` is named Config.xml and resides in the `%SOFTDIAL_ROOT%\smon` folder. The configuration document contains a section for each primary server that the backup server acts as a backup for.

There are many configuration possibilities for `smon`. Rather than discuss in the main document these can be found in Appendix A.

2.5.4. Operating Procedures

Because each backup server requires a certain amount of manual configuration it is also necessary to introduce extra steps in ones operating procedures for commissioning new tenants.

Whenever a new tenant is commissioned, this will result in a number of service instances being created on an application server.

These configuration steps need to be repeated manually on the backup server.

To perform manual installation of a tenant service, run a command prompt as given in Appendix A.

2.5.5. Failing Back onto a Primary Host

Failback is a manual operation. In the event of failover on a permanent host outa of the ge there will be some unscheduled outage of some resources (albeit less than a

minute) while the backup server starts up services and reprovisions those resources.

Failback would have a similar cost and so current practice is to avoid causing the same set of pain to the same set of users twice, and schedule failback as a maintenance activity.

On restart of the primary server, srep recognises that failover has occurred and will not re-establish replication. Instead, srep stops all SCC services in the primary host, raises alerts through CallGem if possible, and in the event log. Srep continues to run and will stop local SCC services from running if started.

Failback to the primary is achieved by the following manual steps:

The general process for failback is as follows:

- Ensure all tenants using failed over service are offline.
- **Stop `smon` on the failover server.**
- **Stop services failed over onto the failover server.**
- **Update configuration back on to the main server. To do this**
- Start services on the main server.
- Validate that service is restored
- Restart `smon` on the failover server

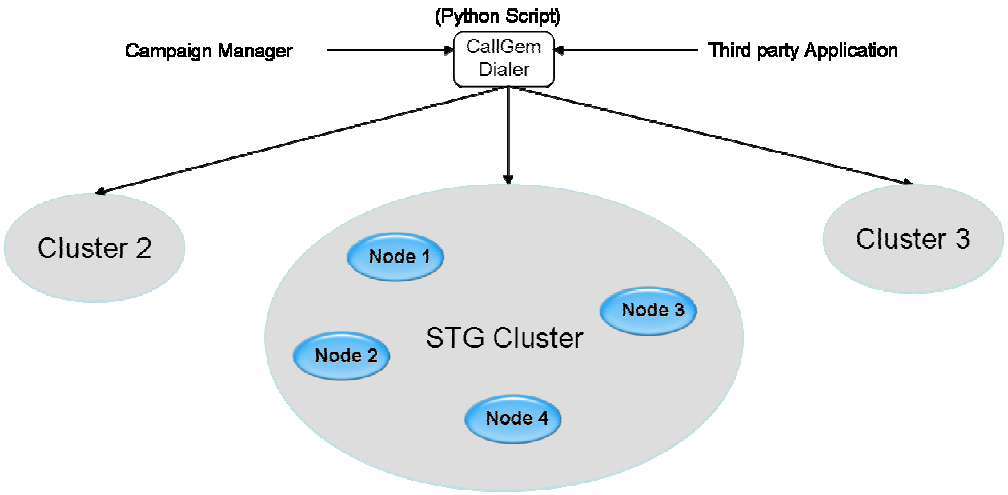
The items highlighted in **bold** should be configured as a batch job to run on the failover server.

2.6 Telephony Clustering

The B2BUA architecture employed by the telephony component of Softdial Contact Center™ does not lend itself to clustering very well.

In order to facilitate clustering it must be possible to bridge any customer call to any agent. STG, with CallGem® acting as a clustering controller, has been designed to accommodate bridging calls between STG instances, and to minimise the requirement to bridge calls.

This means that it is possible to configure multiple instances of STG to behave as a cluster, viz:



Each node in the cluster is a machine or VM running STG. CallGem® can treat the cluster as a single logical telephony node for resource management. Any campaign configured to run against a telephony layer implemented as a cluster may use the resources of all of the nodes in the cluster.
The nodes in each cluster will have to bridge some SIP calls between nodes. The more nodes there are, the number of calls bridging exponentiates. Given default bridging behaviour this limits the maximum number of nodes in a cluster to 9 (8 + 1 redundant).

This leads to a requirement to make decisions on what size each node in a cluster should be. Loss of a node will result in loss of all media (calls on that node) and any calls bridged through that node.

A telephony server instance may carry up to around 1000 channels of voice, which equates to something between 200 and 500 agents depending on activity. However, 'Make it as big as possible' isn't necessarily the best strategy:

Large telephony instance	Small telephony instance
☺ Cluster size of up to 8000 channels	☹ Cluster size limited to 8 * number of channels configured
☺ Reduced management and OS costs	☹ More server instances to manage for a given number of channels.
☹ More channels per node means more voice latency	☺ Lower voice latency with a small number of channels per node.
☹ Requires managed gigabit	☺ Lower bandwidth per instance may

infrastructure or 10GBE if running telephony instances virtualised.	reduce network infrastructure costs and defer need to adopt 10GBE
☹ Increased risk/cost of failure *	☺ Reduced risk/cost of failure *

Note: * Likelihood of failure exponentiates with system size; a 1000 channel telephony server is likely to fail 4 times as often as a 500 channel telephony server.

Rather than have a complex model to determine the optimum telephony server instance size we should set some simple rules for telephony clustering:

1) Any installation delivering more than 300 channels (~100 predictive agents/ ~140 inbound / ~200 IVR endpoints) should implement a cluster to deliver reliability.

2) Any installation delivering more than 600 channels or an installation delivering a managed service to other contact centers should implement a N+1 redundant cluster.

3) Cluster size determines the largest possible campaign. If you are going to run a predictive campaign with 1000 agents you will need a cluster that supports 3000 channels. Dividing the cluster size by 4 gives you a realistic telephony server instance size. If you do not run very large campaigns you should size telephony server instances as follows:

Cluster capacity required	Telephony instance channels
< 1000 channels	500
1000 – 3000	600
3000 – 5000	800
5000 – 8000	1000

2.6.1. Configuring STG Clustering

The STG config file follows the standard Sytel pattern of being located in the %SOFTDIAL_ROOT%\STG folder. In every STG that will be part of a cluster the following elements will need to be added or edited:

```
<clusterMode>true</clusterMode>
<telephonyLayerID>STG_Cluster1</telephonyLayerID>
<telephonyLayerIP>10.7.0.3</telephonyLayerIP>
```

Setting telephonyLayerID - each STG instance within a cluster should have the same telephony layer ID. This is the same telephony layer ID referenced by Campaign Manager when configuring which telephony resources a campaign should use. If there is only one default telephony layer and it is to be implemented as a cluster, the telephony layer ID should be set to '__singlecti'

This tag should be set to the IP address (or the fully qualified DNS name) of the machine to enable successful bridging of the calls within a cluster. This tag is important because it allows various STG nodes in a single cluster to communicate with each other directly, for example in the scenario when the respondent is nailed up using one STG node but the agent is using another.

The only addition to the legacy solution is the addition of a python script that sites next to dialer and allows advanced configurations, in case multiple clusters are used.

To enable multiple STGs to be part of a single cluster:

2.7 Database scaling & resilience

Sytel has selected MongoDB as its database of choice for reporting. MongoDB was developed to offer several advantages over traditional SQL-based RDBMS systems, that make it an ideal fit for Sytel:

- Scaling/ speed – MongoDB facilitates horizontal scaling of the data layer, easier development, and the ability to store order(s) of magnitude more data than was used in the past.
- Flexibility - The document data model (JSON/BSON) is easy to code to, easy to manage (being schema-less), and yields excellent performance by grouping relevant data together internally.
- Power - While some advanced functionality has been left out, much traditional database functionality remains; for example, secondaries indexes, unique key constraints, atomic operations, multi-document updates.
- Ease of use – MongoDB is very easy to install, configure, maintain, and use, providing few configuration options, and trying instead to automatically do the "right thing" whenever possible.

MongoDB has two mechanisms already built in to facilitate deployment in a high availability environment:

1. **sharding** - the method for scaling a system

2. **replication** - the approach for data safety, high availability, and disaster recovery.

2.7.1. **Scalability** via **Sharding**

Sharding is the partitioning of data among multiple machines (cluster) without data duplication. Imagine a database of customers where those with surnames beginning with letters A to F are stored on Server 1, G to R on Server 2, etc.

Applications connect to the sharded cluster through a process called `mongos`, a routing and coordination process that makes the various components of the cluster look like a single system. When receiving client requests, the `mongos` process routes the request to the appropriate server(s) and merges any results to be sent back to the client.

In this way, the sharded MongoDB cluster looks like a single logical server to the application. During periods of high demand, the load is distributed across multiple shard servers.

2.7.1.1 Balancing

Balancing is necessary when the load on any one shard node grows out of proportion with the remaining nodes. In this situation, the data is automatically redistributed to equalize load across shards.

2.7.1.2 Failover

Proper system functioning requires that each logical shard be always online. In practice, this means that each shard consists of more than one machine in a configuration known as a **replica set** (see 2.7.2. below).

2.7.1.3 Shards

Each shard consists of one or more servers and stores data using `mongod` processes (`mongod` being the core MongoDB database process). In a production situation, each shard will consist of multiple servers to ensure availability and automated failover. The set of servers/`mongod` process within the shard comprise a **replica set**.

2.7.2. Redundancy via Replica Sets

Replica sets are a form of asynchronous master/slave replication, adding automatic failover and automatic recovery of member nodes.

A replica set consists of two or more nodes that are copies of each other (i.e. replicas).

The replica set automatically elects a primary (master). Drivers (and `mongos`) can automatically detect when a replica set primary changes and will begin sending writes to the new primary. (The `mongos` sharding process does this too.)

Replica sets provide the following five distinct benefits over the use of a single node.

2.7.2.1 Data Redundancy

Replica sets provide an automated method for storing multiple copies of data.

Supported drivers allow for the control of "write concerns". This allows for writes to be confirmed by multiple nodes before returning a success message to the client.

2.7.2.2 Automated Failover/ High Availability

Replica sets will coordinate to have a single primary in a given set.

Supported drivers will recognize the change of a primary within a replica set.

In most cases, this means that the failure of a primary can be handled by the client without any configuration changes.

A correctly configured replica set basically provides a "hot backup". Recovering from backups is typically very time consuming and can result in data loss. Having an active replica set is generally much faster than working with backups.

2.7.2.3 Distributed Read Load

By default, the primary node of a replica set is accessed for all reads and writes, but it is possible to perform queries on secondaries.

MogoDB provides a method for sharing the read load amongst several nodes.

2.7.2.4 Maintenance

When performing tasks such as upgrades, backups and compaction, it is typically required to remove a node from service.

Replica sets allow for these maintenance tasks to be performed while operating a production system. As long as the production system can withstand the removal of a single node, then it's possible to perform a "rolling" upgrade.

2.7.2.5 Disaster Recovery

Replica sets allows for a "delayed secondary" node. This node can provide a window for recovering from disastrous events such as:

- bad deployments
- dropped tables and collections

2.8 Web Application Scalability and Resilience

For its web applications, Sytel is adopting some well known and widely used patterns for distributed software, the most important of which being use of an application layer (layer 7) load balancer. Because of its high efficiency with HTTP protocol, this is recommended for high-availability and high-performance web applications.

Sytel is currently developing a dedicated load balancer. Until its release, a 3rd party or custom load balancer along the lines described here will suffice.

A layer 7 load balancer acts as a reverse proxy, so it is possible to situate it anywhere in the architecture; e.g. within a sub-network (DMZ), on a server LAN, as default gateway or even in a distant data centre.

A layer 7 load balancer provides higher availability than a network load balancer. This is because a network load balancer is only aware of the availability of a *server*, relying on the ability of a server to reply to a ping or a TCP handshake. An application load balancer, however, can examine the actual *application* layer data available to it, including expected content. This means it will never send requests to an application that has crashed or is offline.

Load balancers can use a wide array of scheduling algorithms, from simple (e.g. random choice, round robin) to sophisticated (e.g. server load, response time, health check, number of active connections, geographic location, capabilities, etc.), but care must be taken if **session persistence** is required (see section immediately below). In particular, the use of 'round robin' in conjunction with persistence-based load balancing should be discouraged. This is because round robin does not consider any factors regarding which instance serves the next

request, and can lead to great imbalance. The smaller the server pool, the more likely this will happen.

2.8.1 Session persistence

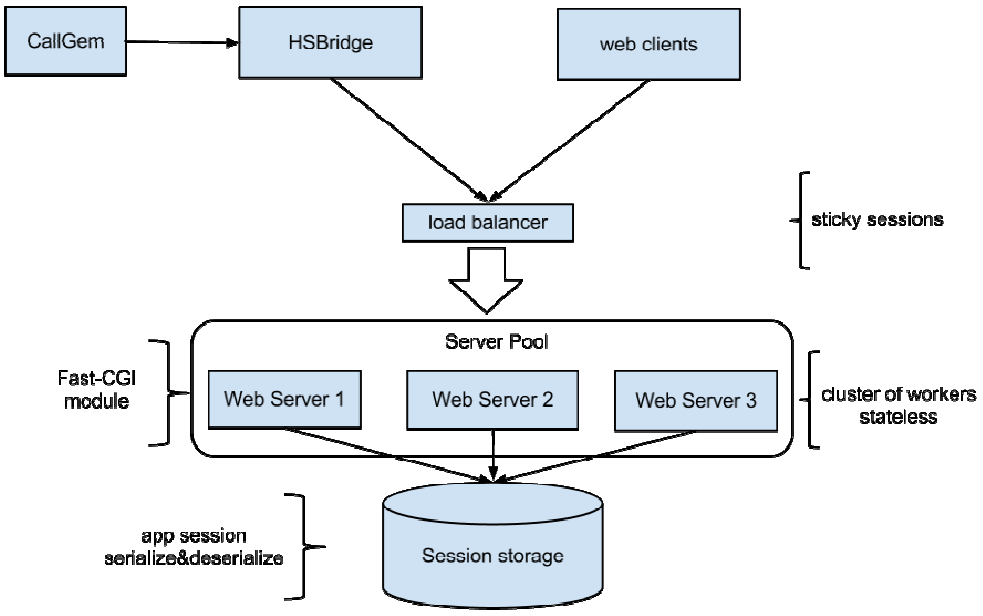
To support session persistence for web applications, load balancers use a mechanism known as 'sticky sessions'. Session persistence is an exception to the rule of load balancing, and takes precedence over the load balancing algorithm. This means that whenever persistence information is present in a request, the load balancer will route the request directly to the server. Persistence is maintained between client and server in a session cookie, using one of two methods:

i. configure the load balancer to set up its own session cookies

ii. use application cookies such as `PHPSESSIONID` or `JSESSIONID`.

2.8.2 Redundancy

Single point of failure can be avoided by configuring two or more load balancers to share the same IP address, with one configured as a failover (i.e. an active/standby pair). Also, the use of a private network whenever possible to connect load balancer to balancer members allows for dedicated bandwidth and offloads SSL handling to the load balancer.



2.9 HTTP tunnelling using NetClient and NetBridge

The core protocol used by SCC services, SDMP, is a TCP-based protocol that is stateful in that resource ownership is determined by the TCP connection that created the resource. This poses challenges for deploying SCC reliably distributed over a WAN. Since WAN links are prone to failure, this can lead to resource failures and outage.

NetBridge and NetClient provide a solution based on HTTP tunnelling and message queuing.

This allows session reliability to be maintained over a network connection that is subject to transient failure.

The http tunnel is also half-duplex and employs long polling for events. This enables firewall traversal without the need for costly VPNs.

Setup of HTTP tunnelling is simple and should be transparent to the user in practice.

2.8.1 Setting up NetBridge

NetBridge is simply a HTTP Proxy for CallGem. It resides on the same server as CallGem and offers a RESTful API to create sessions and send and receive SDMP messages. To deploy, perform the following steps:

- Copy the SNB folder from the HA kit payload to the Softdial Folder on the Controller (CallGem) host.
- Run a command prompt as an administrator with elevated privilege.
- Navigate to the Softdial\SNB directory
- Run the following commands:

```
NetBridge.exe install
net start SNB
```

This will install NetBridge and will start the service listening for HTTP on its default port (87). This can be configured to any port you choose. For configuration options please refer to Appendix D.

2.8.2 Setting up NetClient

NetClient is a service that appears to the user as if it _is_ CallGem. It listens on the same range of TCP ports as CallGem does and consumes and emits SDMP messages in the same way. Every SDMP connection to netbridge is marshalled into a HTTP communication session to a NetBridge instance on a remote host.

NetClient is typically deployed on application servers running Campaign Manager or Scripter, and on third-party servers needing reliable WAN connections to SCC. Instead of Campaign Manager server connecting to CallGem on a remote host, it is configured to connect to the local host on the same port number, and NetClient is configured to connect to CallGem.

To deploy NetClient on an application server perform the following steps:

- Copy the SNC folder from the HA kit payload to the Softdial Folder on the Application Server host.
- Run a command prompt as an administrator with elevated privilege.
- Navigate to the Softdial\SNC directory
- Edit the Config.xml file and update the `<callGemHost>` entry to contain the IP address or fully-qualified DNS name of the host running CallGem
- Run the following commands:
  ```
  NetClient.exe install
  net start SNC
  ```

Lastly, stop any application services on the application server and reconfigure to connect to localhost instead of the remote CallGem host.

For more detailed configuration options please refer to appendix D.

3. Hardware and Network Resilience

This section makes recommendations about how to structure your server environment to deliver a high availability SCC platform. Many of the recommendations require capital investment in server equipment. This is relatively inexpensive compared to outage cost so Sytel's firm recommendation is to do as much of these things as you can (or your IT budget will allow).

3.1 Servers

Virtualisation enables you to minimise your server footprint and makes provisioning new capacity a whole lot easier. Most HA installations will be supported by virtualisation but it is also possible to configure a redundant SCC installation using physical servers.

3.1.1. Virtual Servers

Sytel's view is that any redundant platform ought to be virtualised, with multiple (at least 3) physical hosts delivering SCC services on VMs spread across these hosts. The reason for 3 physical hosts is N+1 redundancy of physical resources. If you require 2 servers worth of capacity having 3 running will enable you to continue to deliver service at full capacity in the event of a physical host failure.

Sytel offers an Excel-based server capacity planning tool as part of the HA deployment kit. This enables you to determine the number and base specification of physical servers required for your virtualisation needs.

3.1.2. Physical Servers

In some computing environments virtualisation may not be practical. In this case you will need to develop your strategy around physical hosts. The unit size of each physical server will need to be small, both physically and from a point of view of resources. Sytel's recommendation at the time of writing is to run hosts with a one or two 4-core processors and 16GB-32GB of memory for the following server roles:

- Controller
- Web Server
- Application Server
- Database Server

Hosts running the telephony server role should be configured with a single 4-core processor and 4GB of memory.

3.1.3. Server Sourcing

When sourcing servers for SCC, having consistent equipment is key to maintaining availability.

For a new installation this means selecting server equipment with a long production run planned. It is often better to buy a smaller upgradeable server that has just been released than a mature offering, on the basis that you will need to source more of the same over time as your installation grows.

SCC is optimised to use as little resource as possible. In practice this means that processor load is not the major limiting factor for server sizing. The major limiting factors in our experience are:

1) Memory
2) Disk IO speed
3) Network IO

When configuring servers (and in particular virtualisation hosts) specifying adequate memory is a priority. Server Memory is usually cheap enough for this not to be a major problem. When specifying memory you need to consider the limits of any hypervisor you are using. At the time of writing, for example, VMWare ESXi (the free version of ESX) limits server memory to 32GB.

A server hosting database or telephony (for call recordings) is going to be subject to huge IO load that needs to be handled in real-time. Capacity on modern systems is not a problem but IO throughput will be. Sytel's firm recommendation is to specify a dedicated hardware RAID controller with as much Non-Volatile cache as you can afford, coupled to either fast SAS drives (15Krpm) or solid-state drives.

If you ensure that you back up VMs religiously, it is worthwhile and sensible to configure hard disks in a stripe set to maximise IO performance.

Virtualisation hosts serving up database, application and voice media traffic can become a network bottleneck.

Any large-scale platform should have physical hosts configured with 10-gigabit Ethernet coupled to a 10-gigabit network switch.

For installations of less than 1000 channels of voice, gigabit Ethernet will suffice.

Servers should be specified with dual NICs as a bare minimum and dual redundant power supplies.

3.2 Managing Resources

When running, servers (whether physical or virtual) have finite memory, processor and disk resources. Consuming all available resource in one of these categories will have fatal consequences for processes on the server instance in question.

If you have a well-defined network management strategy and the tools in place to monitor resources, then this is good news. If not, Sytel has various tools that can be used to provide monitoring capabilities. Since there is no one-size-fits-all approach to network management, a customer setting up a network management infrastructure as part of a high availaibility infrastructure is urged to contact the Sytel support team for a discussion of your needs.

Regardless of whether you have a network management infrastructure in place there are a number of things you can and should do to ensure SCC services do not run into resource problems:

3.2.1. No other applications deployed on SCC hosts

This may seem an obvious requirement but needs to be stated. There are some cases where you will want to install network management software, Antivirus software or 3rd party remote access tools. You must always get agreement from Sytel prior to deployment of such software.

Antivirus software in particular may run privileged code and cause problems with service delivery. The Sytel support team will be able to advise on appropriate configurations.

3.2.2. Separate Log Volume

Log files should always be stored on a separate volume. In this way, impact of a disk full condition is limited to not being able to make further logging rather than causing outage.

3.2.3. Log purge

Log purge cycles are 28 days by default. This should be tailored to ensure that log files would not take up more than 50% of free space even running at maximum capacity.

For example, a controller server serving 500 agents may be consuming 30% of log volume capacity when running a 28 day purge cycle. If the controller is intended to support 2000 agents this would lead to filling up the log volume when running at capacity. In this case the purge cycle should be shortened to 7-10 days so that scaling up will not lead to loss of logging.

3.2.4. Scheduled Service Restarts

Best-practice behaviour for network services is to design and implement the service such that it is either started on demand, or if long-running does not require restart. All SCC services are long-running.

As part of QA Sytel performs load testing and uses instrumentation for leak testing. Memory leaks should not occur. There are however other reasons for having planned service restarts:

1) 32-bit clock. CallGem makes use of a 32-bit clock which counds milliseconds since process start. This clock overflows after 49.7 days with unpredictable results. In order to mitigate the effects of this a weekly scheduled recycle of CallGem is required.

2) Late-binding of plugins and 3rd-party code in Scripter and Workflow Server. Scripter and workflow server both load and invoke 3rd-party code under program control. Users can deploy new versions of scripts and dependent assemblies in real-time. De-referenced assemblies do not get discarded in the same way as other objects. This flexibility of behaviour leads to memory bloat. The only cure for this is to schedule restart of these services. Again we would suggest doing this on a weekly basis.

3.3 Network Infrastructure

A high-availability infrastructure must have no single point of failure.

To this end if you are delivering high availability it means your network backbone connecting together virtualisation hosts, routers and media gateways must be redundant.

The cost of not doing this is complete system failure in the event of outage of a network backbone switch. This is unlikely but not unthinkable.

As a bare minimum your network backbone switches should have dual redundant power supplies. You may wish to consider making these redundant although the IT costs of this are prohibitive

At the time of writing a 10GBE L3 managed backbone switch with 12 or 24 ports and a dual redundant PSU costs between $10,000 and $20,000 depending on manufacturer. Having to replicate a costly piece of equipment with a long MTBF is a matter of judgement for the service provider.

3.4 Power

Your server room (or datacentre) needs at least 2 independent sources of electrical power. Commercial datacentres will provide this as a standard service. The hardware running SCC and the network backbone must be served by 2 separate Power Distribution Units (PDUs) connected to independent electrical sources.

Each device with a redundant power supply must have one input from each PDU.

3.5 Bearer Networks

Configuration to support multiple bearer networks in a HA SCC installation is fraught with complication. In order to provide some simple guidance this document considers two scenarios:

- SIP Telephony service delivered by multiple service providers over multiple WAN interfaces.
- TDM Telephony service delivered by multiple service providers mediated through multiple standard media gateways.

3.5.1. Multiple SIP Service Providers

It is possible to configure STG to use multiple service providers and make selection preferences. STG is limited to authenticating with a single proxy. If your service providers allow you take make calls without authentication you can configure each STG instance to select providers in some order based on your business needs.

If you have more than one service provider and each service provider requires authentication you will need a local proxy. There are several freely-available solutions for this.

- OpenSER is a high volume SIP proxy. It performs the proxy role well, is designed for HA and is used in many carrier environments. However it is not easy to deploy for the novice.
- Asterisk is best known as an IP PBX but can also be used as a SIP proxy. It is relatively easy to configure but does not scale nearly as well as OpenSER.

If using a third-party proxy to route to multiple service providers, STG is configured to deliver SIP to the local proxy. The proxy them makes route determination based on the rules you set and delivers the call to the service provider.

You may already have a SIP server infrastructure based on a SIP server other than the free solutions above. This should not pose a problem. However if your SIP server is B2BUA this may impose some scaling limits.

In a green-field installation Sytel would recommend using OpenSER to deliver a high availaibility proxy.

3.5.2. Multiple TDM Service Providers

SCC is a SIP switching platform so TDM connections are mediated through access gateways.

The first thing to consider when providing a reliable TDM infrastructure is size of failure unit. Consider this scenario; you have 2 different service providers, each delivering one DS3 trunk to you. It would be tempting from a cost perspective to have a single access gateway with a multiple DS3 circuit module manage both trunks. This creates a single point of failure.

In fact, best practice with this topology would be to break out each DS3 to E1 or T1 circuits via a separate CSU, and use smaller (say 8 or 16 E1/T1 span) media gateways to connect a mix of circuits from each carrier.

This approach means:

- a single media gateway failure will result in losing only a portion of the overall capacity from each carrier.
- A complete carrier or CSU failure will result in half of your capacity still being available.

Sytel recommends the following breakout approach for multiple TDM service providers based on line presentation:

Line capacity and carrier presentation	Local breakout	Media Gateways
8-12 E1/T1	E1/T1	4 port E1/T1 gateways
13-32 E1/T1	E1/T1	8 port E1/T1 gateways
1-3 DS3	E1/T1 via CSU	16 port E1/T1 gateway
4+ DS3 or other high bandwidth optical circuit	DS3	DS3 gateways

4. Proving your High-Availability Solution

There is little point in implementing a High Availability solution if you do not perform failure testing before deploying to customers. As can be seen from sections 2 and 3 there is a lot to configure in both hardware and software to achieve high availability. The only way one can be sure that the

system will remain available in the event of a component failure is to emulate that component failure.

Developing a proving plan is the responsibility of the service provider delivering a SCC solution and will depend on what level of resilience you build.

Appendix A – `Smon` Configuration Options

Here is a simple example `smon` configuration:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="sytelco.com/smonConfig">
  <hostConfigurations>
    <hostConfig address="10.0.0.74">
      <startCommands>
        <startCommand command="net start namespace /Y" />
        <startCommand command="net start &quot;Predictive Dialer&quot; /Y" />
      </startCommands>
    </hostConfig>
  </hostConfigurations>
</configuration>
```

Each primary server that the machine is acting as a backup for will require a series of commands configured.

Appendix B – Sample Failback Batch Script

Appendix C – Cluster Configuration Options

Advanced routing of calls through a specific cluster can be configured using the Python script (SelectTelephonyNode.py) in the `%SOFTDIAL_ROOT%\SP\Scripts` folder. This script is used by the dialer service to select a particular node within a particular cluster based on user settings. The following variables are available within this script at run-time:

- Telephone number
- Tenant Name
- Campaign Name
- Agent Extension
- Agent Name
- Telephony Layer ID
- Session ID
- Trunk capacity of a node
- Trunks in use in a node
- Node ID (STG VM's network name)
- Number of nailed up agents

The following are examples of possible modifications using the above variables:

The script by default has access to the trunk information on all nodes and clusters on the dialer. It is therefore possible to route the calls through different nodes/clusters based on current trunk usage, for example.

The default behaviour of the script is to select a node that is consuming the minimum number of trunks in a cluster at any one time. But it is possible to modify the script to wait for a single node to consume x number of trunks first before moving on to a different node within the same cluster. This provides the ability to overflow calls based on pre-defined criteria.

The script can be configured to tie-up interviewer login to a specific node within the cluster. This could be useful in a scenario where a customer would like to tie-up say German agents to a German STG (where the German STG is part of a large cluster of STGs spread across multiple countries). The script provides the ability to log the routing information in the dialer logs for later troubleshooting.

Please note that the script **cannot** be configured in real time; a change in this script requires a restart. Also, please talk to Sytel before making a change to this script, as a complex script can affect the performance of the system.

Appendix D – Tunneling Configuration Options

NetBridge Config.xml options

Option	Default Value	Description
urlStem	http://+:87/NetBridge/	The port and url Path that the NetBridge web server listens for HTTP requests on.
port	6500	The TCP port used for SDMP communication with CallGem. This should not change as a rule.
host	localhost	The host that NetBridge connects to. This should always be the local host but is parameterised in case the loopback address is not accessible, or needs to be specified as a particular IPV4 or IPV6 address
user	Globals._SYSTEM_ACCOUNT	The SCC user that NetBridge sessions authenticate as. Unless you have implemented a security model this should not change.
password	Globals._SYSTEM_ACCOUNT	The password for the SCC user that NetBridge sessions authenticate as.
tenant	Globals._LANDLORD_ID	The tenant of the SCC user that NetBridge sessions authenticate as.

NetClient Config.xml options

Option	Default Value	Description
callGemHost	localhost	The remote host to connect to. This host will run CallGem and NetBridge
netBridgePort	87	The port number used for communicating with NetBridge
listenPort	6500	The base listening port that NetClient listens or SDMP on. Setting 6500 (the default) means that NetClient will listen on ports 6498 thru 6502.

white paper

www.sytelco.com
info@sytelco.com
+44 (0)1296 381 200

Sytel Limited 1 Cromwell Court New Street Aylesbury Buckinghamshire HP20 2PB UK