ARCHITECTURE GUIDE

Campaign Manager 6.0

VERSION CONTROL

Version	Date	Author	Changes
1.0	28 April 2017	D Cooper	Release

RELATED DOCUMENTS

The related documents are located in the Alterian product help.

Name
Campaign Manager 6.0 Installation Guide
Campaign Manager 6.0 Upgrade Guide
Campaign Manager 6.0 Backup Guide
Campaign Manager 6.0 Data Flow and Structure
Campaign Manager 6.0 Load Process Guide
Campaign Manager 6.0 Release Note

TABLE OF CONTENTS

1. INTRODUCTION

This document provides an overview of common architecture configurations for a Campaign Manager 6.0 system. It includes guidance on security and firewall configurations, and information on how to plan the physical hardware required for your environment.

This document is intended for those who are responsible for planning or installing a Campaign Manager environment for the purposes of hosting the application.

For detailed installation instructions, please see the Campaign Manager 6.0 Installation Guide for Windows Server 2008 R2 SP1 and Windows 2012 R2 Guide or the Campaign Manager 6.0 Upgrade Guide.

Please review the related documents listed on Page 2 for other information that may be useful in planning your environment.

If you require further assistance, contact your Alterian account representative or Alterian Support via support@alterian.com.

2. ARCHITECTURE OVERVIEW

2.1. TECHNOLOGY STACK

Campaign Manager is built predominantly on a Microsoft technology stack that delivers an n-Tier application via the internet.

Please refer to the Campaign Manager 6.0 Installation Guide or Upgrade Guide for supported platform details.

Do not perform upgrades to newer versions of SQL Server and Silverlight without consulting the latest release notes for supported versions.

The Installation and Upgrade Guides for each release will outline supported versions and pre-requisites for the technology stack and inter-application pre-requisites and should always be referred to prior to commencing a project.

2.1.1 LOGICAL ARCHITECTURE

The logical architecture for Campaign Manager is pictured in Figure 1, below. The recommended approach is to place the App Server, SQL Server and Engine Server roles all within the trusted network, and direct traffic to the App Server through the firewall by means of a reverse proxy. This is the most secure way to make the application publicly available over the web, and yet support the Microsoft DCOM client requirement on the App Server, which is the communication protocol for Engine.

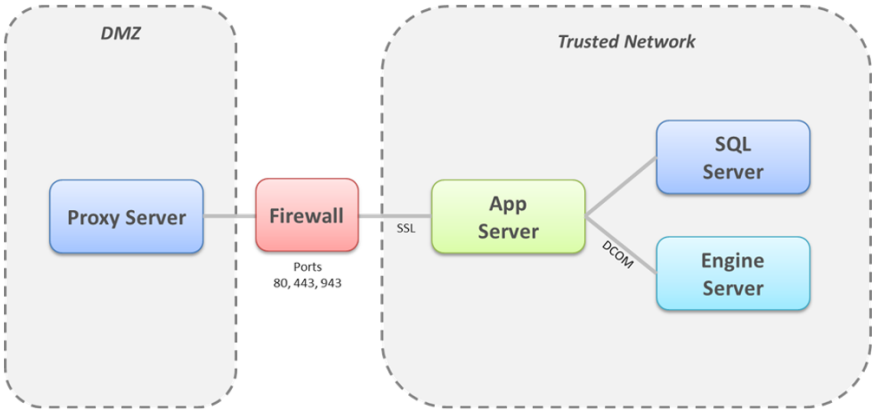


Figure 1 - Campaign Manager Logical Architecture

2.2. SERVER ROLES DEFINED

The following server roles are required for each installed Campaign Manager environment. A role is not necessarily a physical machine, but a major collection of components that have specific application requirements on that host machine.

It is assumed that each of the machines in the trusted network reside on the same network domain, and are physically located in close proximity to one another.

Application Server – The role of the application server is to host the web application and services for Campaign Manager.

SQL Server – The role of the SQL server is to store Campaign documents i.e. campaigns, segments etc. and metadata such as users, permissions etc.

Engine Server – The role of the Engine server is to run the Engine database hosting data for the Customer Data Mart, Campaign Manager State Data and Campaign Manager History Data.

2.3. APPLICATION SERVER ROLE

2.3.1 SUMMARY OF REQUIRED COMPONENTS:

Windows
.Net Framework 4.5.2
Web Applications (See below for more detail)
Hosting
Services
Authentication
Alterian Engine **Windows Services (See below for more detail)**
Alchemy AlertLogger
Alchemy AlertNotifier
Alchemy CMListenerService *Installed but no longer used*
Alchemy DocumentScheduleService
Https Certificates where appropriate

The App Server hosts three web applications, under a single website.

Here is a summary of each:

1. Hosting: This is the 'gateway' to the system. In a multi-client environment, clients will share a hosting site each accessing it via a 'client specific' host header although it is possible for each client to have their own hosting it is not the norm.
2. Services: The 'workhorse' application, which either re-directs or performs the bulk of the processing requested by each user's interaction with the system.

3. Authentication: As its name suggests, it handles the request to access resources from the Campaign Manager service itself or Email Manager service if that has been included in the solution. Certificates are used to validate these requests.

Note. Engine will be installed on the App Server to facilitate DCOM communication between it and the Engine server. It is not necessary to have an Engine repository on this server, and a license is not required. This installation is carried out within the Campaign Manager Deployer.

This App Server role hosts .NET assemblies for the application middle tier and serves content to all "client" computers using Windows Server IIS. The Campaign Manager website communicates with the client computer's browser, including the Silverlight plugin over either HTTP or HTTPS.

Note. If the Campaign Manager system is using the integrated Email Manager system then they must share the same protocol i.e. ALL http or ALL https.

2.3.2 APP SERVER SERVICE TABLE

A description of each service is as follows:

Name	Purpose	Default State
Alchemy AlertLogger	Takes the .txt files from the Alerts folder and puts their contents into AlMain.AL.alertlog table.	Automatic
Alchemy AlertNotifier	Reads AlMain.AL.alertlog table and sends out alerts according to its configuration.	Automatic
Alchemy CMListenerService	Currently not in use.	Automatic
Alchemy DocumentScheduleService	The main service handling campaign execution. The service manages: • Track and TacticOutput execution. • Audience creation and update. • Event and Trigger processing. • Scheduled document execution.	Automatic

2.4. SQL SERVER ROLE

2.4.1 SUMMARY OF REQUIRED COMPONENTS:

Windows
.Net Framework 4.5.2
SQL Server
Database Engine Services
Full Text Search
Management Tools
File Services
SQL Database Engine (See below for more detail)
3 'System' Databases
AlMain
AlterianAuth
AlchemyStore
Client Databases (CM) – one per hosted client
Windows Services
Alchemy AlertLogger
Alchemy AlertNotifier
Alchemy SQLJobService

This server role hosts three system databases to handle authentication, management and storage of application metadata. A separate client database is required for persistence of user created content.

The three system level databases which are shared across all hosted clients are:

AlMain: This database holds system settings such as Time Zone information and the tables used to issue alerts.

AlterianAuth: This is the credential store for the authentication server. It stores user logins and a mapping to which applications those users have permission to use.

AlchemyStore: This database is no longer used.

2.4.2 SQL SERVER SERVICE TABLE

Here is a description of each service.

Name	Purpose	Default State
Alchemy AlertLogger	Takes the .txt files from the Alerts folder and puts their contents into AlMain.AL.alertlog table.	Automatic
Alchemy AlertNotifier	Reads AlMain.AL.alertlog table and sends out alerts according to its configuration.	Automatic
Alchemy SQLJobservice	The main service handling campaign execution. The service manages: • Document scheduling. • Campaign Track and Event scheduling • Maintenance tasks.	Automatic

2.4.3 ALCHEMY (OR CLIENT NAME) DATABASE

Campaign Manager architecture supports the creation and management of multiple Client DB's to securely exist within a single software environment to facilitate multiple clients being hosted on a single SQL server. The "Alchemy" Client DB is the default client database created by the Campaign Manager Deployer, subsequent Client DB's are created via a provided SQL script or new client API, and these databases are used to store client specific data. There will be one of these per hosted client within the system and the names of subsequent Clients DB's will be set on creation. The data held in this database corresponds to Campaign Manager's working data: users, documents, templates, folders, tools, groups, and permissions.

More information on the creation of Client DB's is available in the Campaign Manager Administration Guide.

It is important to note that user activity on Campaign Manager places minimal load on SQL Server for all analytics and outbound message creation and execution.

In an environment configuration where the SQL Server is not installed on a stand-alone machine, it is important to control SQL Server's memory usage to ensure the rest of the machine has enough ram to operate. Please note that even when installed on its own machine, SQL Server's memory usage should be in line with Microsoft's best practice guides to ensure the OS has enough operating RAM to function correctly.

Guidance on minimum memory requirements SQL Servers on the Microsoft MSDN site; for reference, here is an example article on setting maximum memory for SQL Server:

http://technet.microsoft.com/en-us/library/ms177455(v=sql.105).aspx

2.5. ENGINE SERVER ROLE

2.5.1 SUMMARY OF REQUIRED COMPONENTS:

Windows Server
Net Framework 4.5.2
Alterian Engine **Windows Services (See below for more detail)**
NucBroker
AltDocumentcache
AltLogService
Alchemy AlertLogger
Alchemy EngineService
Alchemy AlertNotifier

This Engine Server role hosts the Engine database application. Campaign Manager 6.0 installs Engine as part of the Deployer. Each Campaign Manager account can support multiple Engine Projects although best practice would be to have one Datasource for easier management and reporting across multiple campaign keys.

2.5.2 ENGINE SERVER SERVICE TABLE

A description of each service is as follows:

Name	Used By	Purpose	Default State
NucBroker	Engine	Manage client application connections to Engine tenants *Previously called ConnectionBroker	Automatic
AltDocumentCache	Engine	Caching of crosstab result sets	Automatic
AltLogService	Engine Surrogate	Asynchronous logging of NucleusSurrogate messages.	Automatic
Alchemy AlertLogger	CM	Picks up alerts created by EngineService and writes them to SQL Server.	Automatic

Name	Used By	Purpose	Default State
Alchemy EngineService	CM	The main service handling campaign execution. The service manages: • Transfer to Engine server and subsequent management of Campaign History Data files. • Creation of staging tables of Contact and Response data. • Archival of staging tables into Campaign Manager History Data. • Updates of Master Campaign table. • Datasource object refresh *For more information Data Flow refer Campaign Manager 2016 R1 Data Flow and Structure.	Automatic
Alchemy AlertNotifier	CM	Provides redundancy for AlertNotifier running on other servers.	Automatic

2.6. PROXY SERVER ROLE

This server role is optional, but highly recommended so that a reverse proxy can protect the DCOM enabled App Server from punching a large range of holes in outside firewall, making all requests from the outside look as though it originated from the reverse proxy itself. Any Linux box with Apache or Microsoft ISA Server or the Threat Management Gateway product would be suitable. A load balancer could achieve a similar objective. This document assumes a device is already in place, which can be used for this purpose.

2.7. STMP SERVER

This server is highly recommended so that internal alerts can be emailed to system administrators about the health of the Campaign Manager system. It is an optional server role, not included in the main diagram above, but it provides good feedback on unexpected environment conditions. This document assumes a machine is already in place, which can be used for this purpose.

2.8. ENGINE MULTI-TENANCY

Multi-Tenancy allows multiple client Projects to run on the same Engine server. Each Project is completely segregated and has its own Engine Repository and Engine process.

This removes the need to buy and maintain a separate Engine server for each customer solution, where those customer solutions are sufficiently small that there is no risk of resource contention (either memory or hard disk) in running the solutions on the same server. This means that an Engine server will be securely hosting multiple Engine projects for multiple clients.

Highlights:

- Multiple Engine Projects may be securely hosted on single piece of hardware
- Sharing hardware may reduce the total cost of ownership
- Hosting multiple clients repositories on a single physical box may result in better performance than virtualization
- Multi-tenancy is only suitable for smaller databases or low concurrency accounts

For further information, please consult your Alterian account representative.

2.9. SPLIT CUSTOMER DATA MART AND CAMPAIGN MANAGER STATE DATA

As well as housing the Customer Data Mart database, the Campaign Manager system holds and maintains Campaign Manager State and Campaign Manager History Data in Engine which, based on the number of active campaigns and the volume of daily activity, can be quite significant.

The Customer Data Mart can be separated from the other two system databases at an Engine level. This has a number of performance and architecture advantages.

Highlights:

- Splitting Customer Data Mart and Campaign Manager system data allows for an A/B Repository switch method to be implemented in the overnight load process to minimize system downtime due to Engine load processes
- Increases disk I/O by allowing configuration put the two repositories on different physical disks to limit disk I/O bottlenecks.
- Logical separation of Customer Data Mart and the applications Campaign Manager State Data for backups etc.

This subject is covered in more detail in the Best Practice Guide - Minimizing downtime with A/B Repository switching.

2.10. SHARED REPOSITORIES

Campaign Manager 6.0, releasing Campaign Manager with Engine, is not tested or supported with an Engine configuration that utilizes Shared Repositories, but the addition

of support for this setup is on the roadmap for a future release. The Alterian Product Management team would be very happy to discuss the use of Shared Repositories with any Customers wishing to use it within a Campaign Manager environment. Please contact your Alterian Account Representative to facilitate that discussion.

3. SECURITY CONTROL

3.1. AUTHENTICATION SYSTEM

The SQL Server database 'AlterianAuth' contains all the credentials that each user will authenticate against for that particular SQL instance. At the current time, each instance must have its own Auth server/database. To actually setup logins and passwords, see the Administration section of this document.

A named user is explicitly allowed only one active session at a time. Subsequent logins boot the prior session out. The session token is only returned when authentication is successful for either of the authentication methods detailed below. Session tokens are specific to IP, user name, client account, and length of time. Any attempt to modify a session token parameter to gain access to another account or other mischievous purpose will invalidate it going forward.

3.1.1 FORMS BASED LOGON

Campaign Manager is typically configured to allow users to enter a login ID and password to gain access to the system. This information is stored in a table named AL.LoginIndex in the AlterianAuth SQL database. The password is securely stored as a salted hash of the actual password.

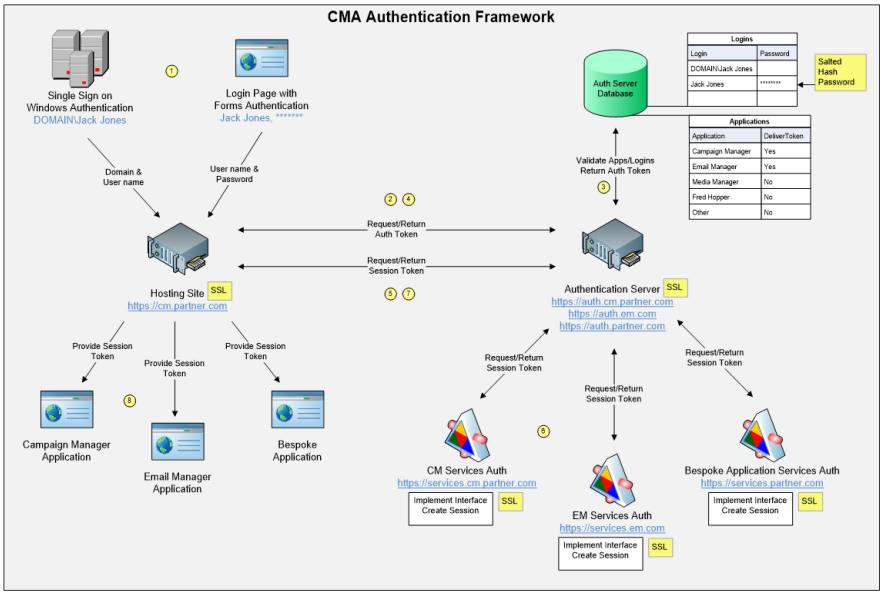
3.1.2 SINGLE SIGN ON

Configuration of Single Sign-On is documented in the Campaign Manager 2016 R1 Administration Guide.

3.1.3 AUTHENTICATION FRAMEWORK

The Authentication Framework is an extensible framework that allows other third party applications to authenticate into Campaign Manager for the purposes of seamless integration by implementing a simple interface to create a session token that can be used to securely have trusted applications communicate over the internet. See Fig. 3 below.

Session inactivity timeouts are supported so that an inactive session will be terminated.



3.2. ACTIVITY LOGGING

Campaign Manager logs valid and invalid login attempts, including the invalid login attempt reasons. Additional logging and a user audit trail is due to be included in a future version. This includes logging usage of any entity which is securable in the user interface, for example, where a user has permission to output data, the log would include detail like when, who, how long, to where, file names, etc.

Document usage is recorded today with a time/date and user stamp on every modified record.

All activity logs are stored in SQL Server and can be secured in the same manner as other SQL data. A user cannot use the user interface to view log information generated by the activity of other users.

3.3. NOTIFICATION SYSTEM

The Campaign Manager system has a service that is registered on each server named Alert Notifier. This service takes information\warning\error messages that are logged by the applications in the ALMAIN.AL.Alertlog and sends those out as emails to various people configured in the system. The tables associated with the notification system are described below:

AL.notifyindex	Contains the email addresses to which the alert system sends emails
al.notifycategoryindex al.notifygroup	These two tables are used to create groups for notifications based upon notifyindex.
al.alerttypeindex	Alert types which are logged by the system

AL.alertnotify	Groups to which the types of alerts from alerttypeindex should be sent.

3.4. DATA ENCRYPTION

3.4.1 SQL SERVER

The SQL encryption feature (available on the Enterprise level) to enable rest encryption can be used to protect sensitive data on the SQL Server. As typically encountered, using this type of encryption could have a negative performance aspect. Consult the following Microsoft knowledge article to determine if it is right for your deployment.

http://msdn.microsoft.com/en-us/library/bb934049.aspx.

The Engine server does not support a native encryption on disk feature, but being a column based system it is virtually impossible for users to gain insight into customer specific data just because of the separation of rows of data into separate columns that have different indexing and storage strategies. If greater data security is required, a hardware based encryption system on a SAN or other high-end disk array will usually provide features to encrypt data that just sits on the disk. As with SQL Server, these mechanisms provide added security but usually come with a cost of performance. To see which approach is right for you conduct a trial prior to rolling into production. Faster CPU's or faster disks may be able to offset the performance penalty of encryption.

4. ENVIRONMENT ADMINISTRATION

Automatic Windows updates can affect the performance of Campaign Manager. In particular, they have been observed in some cases to cause degradation of Campaign output performance. We would encourage system administrators to follow Microsoft best practice guidelines with regards to Windows updates, but to also keep in mind the production usage patterns of the system that they are administering.

Administrators should take every effort to avoid unplanned restarts of Engine. This has particular implications for campaigns which are currently processing. See the Campaign Manager Administration Guide for a section on 'Managing Engine Restarts'.

5. FIREWALL CONFIGURATION

The Campaign Manager Deployer utility makes extensive use of Windows Management Instrumentation (WMI) to communicate between servers during the installation. Where there are firewalls between servers, this may interfere with communication. This section provides best practice recommendations for environments where firewalls exist between the physical Web Server and other physical backend servers. It provides guidance on what ports need to be opened to allow communications between the various servers. Note that these are recommendations only, and will depend on what ports are available in your environment.

5.1. EXTERNAL FIREWALL

The external firewall facing the DMZ should be configured to allow port 80 and 443, to allow HTTP and HTTPS respectively. One additional port 943 is required for Silverlight.

5.2. BETWEEN APP SERVER AND ENGINE SERVER

DCOM

Engine requires communication over DCOM if the App Server and Engine Server roles are installed on separate hardware. If a firewall is placed between the App Server and Engine Server the following considerations should be observed to ensure DCOM communication can succeed.

DCOM uses dynamically assigned ports in the range 1025-65535 for all method calls and callbacks. In order to set up DCOM to work with firewalls, the range of ports that DCOM can use should be restricted to reduce the number of ports needed to be opened. As a callback might be on a different port than the method call, the range should be configured on both machines.

The following steps show how to set up an example DCOM port range of 5500-5700. Best practice recommendations from Microsoft specify a minimum range of 100 ports above 5000. See http://support.microsoft.com/kb/217351 for more information.

Follow these steps on both the App Server and the Engine Server:

1. From the **Start** menu, click **Run**. Type in dcomcnfg, then click **OK**.
2. Expand **Component Services** then expand **Computers**.
3. Right-click **My Computer** and select **Properties**.
4. Click the **Default Protocols** tab.
5. Ensure that Connection-oriented TCP/IP is present and top of the list. Click to select it.
6. Click **Properties**.
7. Click **Add**.

8. In the port range field type in 5500-5700. Click **OK** to close the Add Port Range dialog.
9. Click **OK** to close the Properties for COM Internet Services dialog.
10. Click **Apply** on the My Computer Properties dialog.

The firewall should then be configured to allow bi-directional traffic on the specified range of ports.

5.3. BETWEEN APP SERVER AND SQL SERVER

SQL

If a firewall is placed between the App Server and SQL Server, open the port 1433 for bi-directional access, which is required for standard Microsoft SQL Server connections.

UNC

The App Server will require access to SQL server to write file streams over ports 135 through 139, including port 445 which allows file sharing between the web and the managed file system of SQL Server. All of the test ports should be configured as bi-directional.

5.4. BETWEEN ENGINE AND SQL SERVER

If a firewall is placed between the Engine Server and SQL Server, open the port TCP 1433 for bi-directional access, which is required for standard Microsoft SQL Server connections.

5.5. BETWEEN APP SERVER AND SMTP SERVER

If a firewall is placed between the App Server and SMTP server, open port 25 for bi-directional access, which is required for to send notification emails from the AlertNotifier service.

6. PHYSICAL HARDWARE PLANNING

Physical hardware planning is an installation specific process that involves discussion on many variables to come to an educated decision.

At a high level, each server role will perform a series of tasks which will utilize the hardware on the machines as follows:

- App Server – This server role is very process based using CPU and RAM to support more users on the system. The ley is to balance the CPU/RAM usage
- SQL Server – Storage of campaign documents, security, metadata etc. and the access of these is the role of this server
- Engine Server – The database server hosting the Customer Data Mart, Campaign Manager State Data and Campaign Manager History Data. High volume of disk space will be required as a starter and this is the disk space where most of the data expansion will take place in the future. The Engine processing both use based and system based will also use CPU and RAM so both should also be high specification.

The following items can all have an impact on the server specifications:

- Architecture – Single box, single client deployments give dedicated hardware and reduced network traffic but must be of a high enough spec to cope with spikes of performance. Multi-box, multi-client solutions give advantages of shared hardware but will introduce more network traffic.
- Database Size – The size of the Customer Data Mart is one consideration but is only part of the story. Campaign volumes will increase the size of the Campaign History which, when reported on, can often be the produce the highest memory and CPU usage spikes
- Campaign Volumes –The specification for a system running 300-400 campaigns per week would be vastly different to running just 10-20 and but then a further consideration is the volume of contacts per campaign,
- Concurrent Users – Another important consideration, not only for concurrency of users but also the usage profile of users. Having 5 users all using the reporting and analytics functionality is going to produce some high spikes of CPU and memory usage that need to be catered for, whereas if the user are predominantly campaign users, this might not be quite as much impact.

The list above are all component parts to producing a picture of the proposed system, to ensure a performance for the users that meets their needs and also the reason why trying to define a system based on single attributes, such as number of customers in the database, has not proved to be a reliable or accurate specification estimation tool. The Alterian sales process will allow the best forum to discuss hardware specifications when

all of the above items can be discussed with our consultants to ensure the hardware is purchased to meet the need on a case by case basis.