EASYVISTA

White Paper Service Manager - On Premise

TABLE OF CONTENTS

Contents

Download the PDF file

INTRODUCTION

The purpose of this white paper is to help you understand how EV Service Manager can be integrated into your technical environment.

Because individual constraints and technology choices make each client's infrastructure unique, every project will undergo a specific analysis during the pre-sales and/or installation phases.

PRODUCT GLOSSARY

EV Service Manager is based on:

- A **Front Office service** (Service Apps): Provision of a configurable services portal to your end users.
- A **Back Office service** (Service Engine): Provision of a more comprehensive product interface to your Back Office team in charge of dealing with incidents, changes, etc.

Depending on your project and how work is allocated between your end users (Portal) and your technical teams, either the Front Office or the Back Office part can dominate. The target architecture and integration into your infrastructure must take this allocation into account.

OVERALL ARCHITECTURE

Components of different services

Service Engine

- **Web Front End**: In charge of processing http requests from users and returning HTML webpages.
- **Application Server**: In charge of processing business requests and providing the necessary data to the web server while taking into consideration the connected account and its authorizations.
- **Database Server**: In charge of storing data.

Service Apps

- **Web Front End**:
  - In charge of processing http requests from users and returning HTML webpages.
  - It interfaces with Service Engine resources: the web front end server(s) and the application server(s).

Adaptability to your constraints

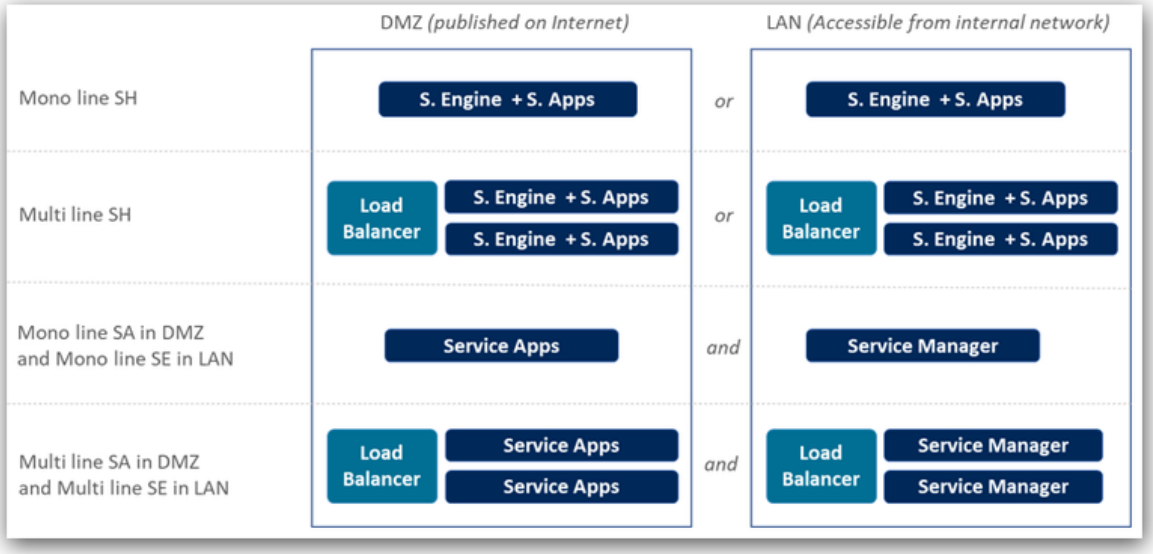
Progressive scaling

The EV Service Manager architecture is scalable. It can be reviewed and modified based on changes in your requirements.

You can start your project with a basic architectural model and review it subsequently if the number of concurrent users increases, if your security rules change or if functionalities are added to the initial project scope.

Each tier can be scaled separately using more or less resources based on the requirements identified.

The diagram below gives examples of possible architectures (positioning of web front end servers only: application servers and databases are usually found on LAN).



Scalability

Our services can go from simple platforms with two servers (one web server and one application/SQL server) to configurations comprising several dozen lines, potentially split into different security zones.

The first design criterion relates to scalability, the aim being to use the two dimensions available to you:

- **Scale IN**: Addition of CPU or memory resources to existing machines so they can handle a greater workload. While this approach is cost-effective (fewer VMs to manage, fewer licenses, etc.), it quickly reaches its limits because certain internal system resources are not expandable (thread, etc.).
- **Scale OUT**: Addition of new machines to take some of the workload. This is the most flexible solution but involves complicating the architecture by incorporating a Load Balancer to divide users between different servers, a filer to share resources, etc.

Regardless of the architecture, and without constraints other than scalability, it is always worth using SCALE IN as far as possible to add new machines.

Resilience

What is the platform's desired availability level? A high level (24/7, for example) will involve a more complex architecture that includes additional lines based on the desired scalability conditions.

These additional platforms will take on the workload in the event of failure of one of the base lines defined as necessary for handling the target workload.

The technical solution will depend on the tolerance that you desire and, therefore, the degradation in service or the complete temporary loss of this or total loss. Here are some examples:

- Temporary loss of an EasyVista line (web and/or application): Add an addition line.
- Temporary loss of a database server: The database server must be clustered for greater availability. But if a loss of several hours is tolerable, VM restart and database restore can be sufficient.
- Temporary loss of the Load Balancer: Load Balancer clustering.

We recommend that you permanently integrate these additional resources into the production chain rather than keep them offline and ready to be started. Although the cost of a running machine is greater than the cost of an offline machine, this will ensure that these machines are correctly configured / up to date when you need them.

Maintainability

If your security policy requires that you regularly update operating systems, we recommend integrating an additional line to the line which has already been sized for scalability.

This additional line, which is not included in the workload expected, will allow you to update your operating systems without impacting production.

Here's an example with three lines, A, B et C (C having been added while only A and B are required for the workload):

- A and B in production; C taken out of production + updated + returned to production.
- A and C in production; B taken out of production + updated + returned to production.
- B and C in production; A taken out of production + updated + returned to production.

Two lines are available at each step, ensuring that users are not penalized.

Segregation of Front Office / Back Office access

To meet your security needs, the Front Office and Back Office tiers can be separated, allowing you to, for example, position one or more Front Office lines in the DMZ (so that they're accessible from outside your network), with the Back Office lines (or even certain Front Office lines) placed on the LAN (for access by your internal teams).

Remember, unless they connect to your network via a VPN, users of mobile applications produced with Service Apps will be located outside your LAN and will have to go through the DMZ to access the application.

Load Balancer

Our services can be placed behind a Load Balancer to evenly distribute users wishing to connect to the different lines available.

Your Load Balancer must allow "session persistence", in other words, a user that has already been authenticated by one of the lines must be directed to that line for as long as they are connected. If not, the user's previous authentication will not be found and the service will ask them to log in again.

Reverse proxy

Our services can be placed behind a reverse proxy. The reverse proxy can, among other things, change the type of request (incoming https to http on the web front end), but it must not:

- Change past parameters.
- Change the URL itself (by adding folders, changing the domain, etc.).

If the reverse proxy includes content control functionality (WAF, antivirus, etc.), it must not change the content passing through it (parameters, content, etc.) apart from its own HEADERS parameters.

Security of your data in transit

To safeguard the confidentiality and integrity of data flows, we strongly recommend protecting Service Apps and Service Engine web front end servers with SSL certificates.

We also advise the following:

- Use certificates provided by a trusted third party. Private certificates will work but will generate lots of errors when accessed from mobile devices outside your network (mobile phones, etc.).
- Configure your SSL so that it does not accept protocols, ciphers, etc. that are known to be vulnerable:
    - SSL v2, SSL v3, TLS v1.0, TLS v1.1.
    - RC4, 3DES, etc.

Here's an example that you can use to configure your SSL.

```
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLCompression off
SSLCipherSuite  ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK
```

Need for other environments

As well as the production environment, you can create additional environments to meet your organization's needs.

We recommend creating and maintaining at least one additional environment so that you can test changes before applying them to your production environment, in particular:

- Fixes or major changes to our products.
- Configuration changes to server components (Apache, PHP, SSL, etc.).
- Version upgrades or fixes to operating systems or server components.

Browsers

Suppliers

The browser market is constantly evolving, so please refer to our Supported browsers wiki page for an up-to-date list of compatible browsers.

Configuration

Pop-ups and JavaScript must be enabled and authorized for Service Apps.

The limit of the local cache and temp files must be adequate (> 10 MB).

If you are using the SSL protocol, you should check that the cache is authorized for the secure page.

Antivirus

On the client workstation, the local antivirus software should not check .JS (JavaScript) files systematically because this can lead to performance issues when displaying pages.

Miscellaneous

Our services do not require APPLET or ActiveX on the client browser.

Cookies

Our services use cookies to improve website functionalities and user experience. These cookies do not contain personal or sensitive data.

Your browser must authorize our services to create cookies.

Format of URLs

Four URLs must be available for each platform:

- URL for Service Engine (Back Office).
- URL for the Demonstration/Training Environment tenant (Front Office) for end users.
- URL for the Production Environment tenant (Front Office) for end users.
- URL for the Sandbox Environment tenant (test) (Front Office) for end users.

The XXXX root of the URL, https://XXXX.yyyy.domain must be unique for each platform and correspond to Apache virtual host incoming streams on the web front ends.

Accessible storage area between servers

Storage area shared between web servers

Multi-line architectures require a file storage area that is accessible to the different web servers so that they can share common files (uploaded resources, styles, etc.).

Symbolic links are created on the web servers to point towards the shared NFS (4.0 and higher) resources on the filer.

Area for exchanging files between the Service Engine web front end and the Service Engine application server

This area is used for data integration processes.

In mono-line architectures, it usually takes the form of a folder on the Service Engine web front end server (SAMBA).

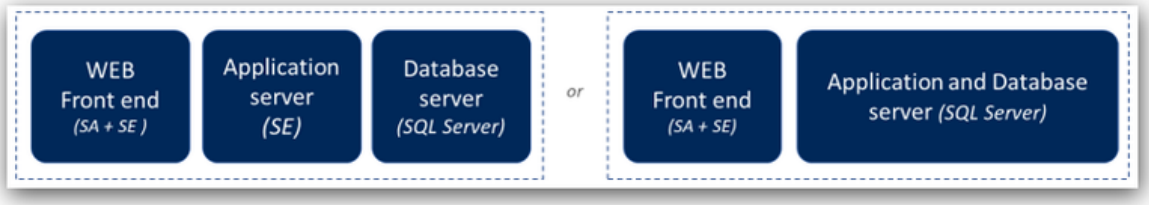
In multi-line architectures, it is found on the filer shared between the different web servers (SAMBA).

ARCHITECTURE EXAMPLES

Mono-line architecture on a LAN or in a DMZ

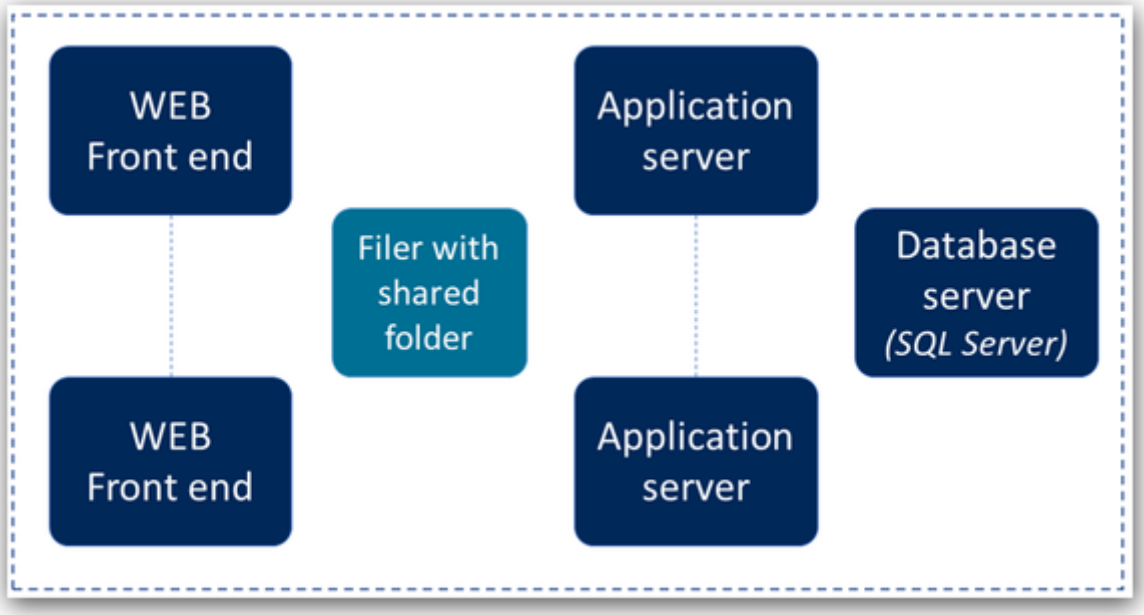
A single line reduces hosting costs but does not guarantee maximum availability.

The "Application server" and "Database server" can be grouped together on the same machine by allocating available resources (CPU, RAM, hard disk) accordingly.



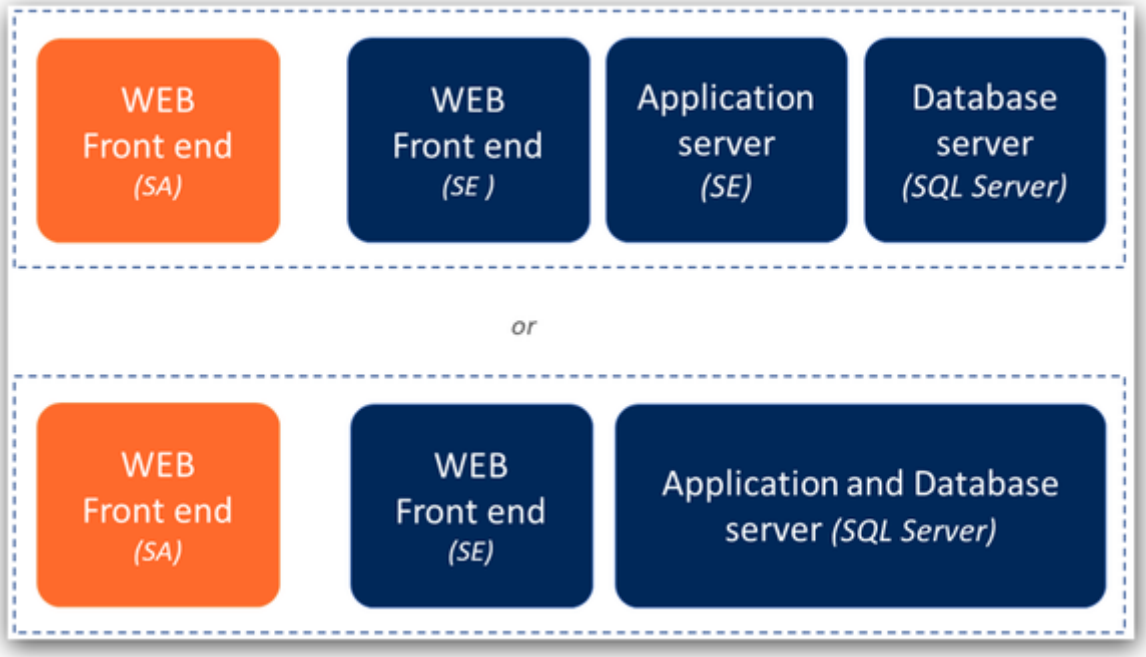
Multi-line architecture

Multi-line architecture offers scalability (number of base lines) and high availability (one extra line for an equivalent service, even losing one of the lines).



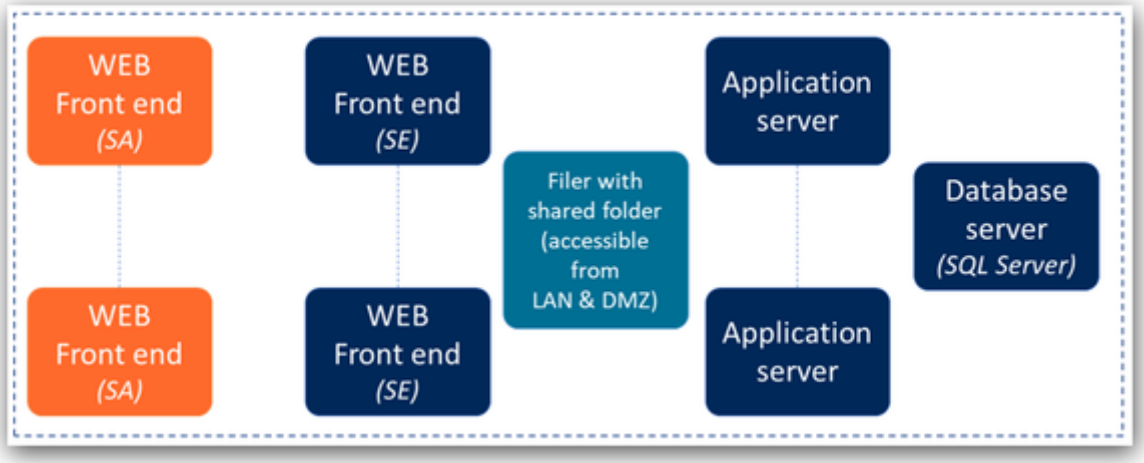
Mono-line architecture with Front Office in the DMZ and Back Office on the LAN

This architecture improves security by only authorizing Back Office access to users connected to the corporate network when access to the portal is available from the Internet (potentially with restrictions by IP, VPN, etc.).



Multi-line architecture with Front Office in the DMZ and Back Office on the LAN

This architecture combines security and availability by only authorizing Back Office access to users connected to the corporate network when access to the portal is available from the Internet (potentially with restrictions by IP, VPN, etc.).



TECHNICAL INTEROPERABILITY

REST

REST and SOAP 1.2 provider

Our services are accessible as REST and SOAP 1.2.

REST and SOAP 1.2 client

The services can use external REST or SOAP 1.2 services.

Email

Sending emails

Service Engine and Service Apps must access your email server to send emails to your users when handling incidents / changes.

The following protocols are supported: SMTP / SMTPS / SMTPS with TLS.

Automatic creation of tickets via email

Service Engine must be able to access working email inboxes to which your users can send messages that will be automatically turned into tickets by the application.

The following protocols are supported:

- POP3 / POP3S.
- IMAP4 / IMAP4S / IMAP + TLS / IMAPS + TLS.
- Office 365 corresponds to IMAP4 in modern authentication (XOAUTH2).
- Microsoft Graph using Outlook REST API with modern OAUTH2.0 HTTPS authentication.

SERVER CPU AND RAM REQUIREMENTS

Caution: These figures are provided as a guide only because the resources required will vary depending not just on the number of users, but the number of incidents/requests created daily, web service activity and the type of use (Front Office, Back Office). The figures provided are based on our SaaS experience.

Remember, your target architecture will comprise one or more of the servers described below.

Web Front End

The requirements are given for **100 Back Office** users at peak usage so as to achieve optimum service quality.

The server is dedicated to the Service Engine application and cannot be shared.

A minimum figure is also given and represents the minimum resources to deploy, even if the number of users is far below 100.

Use case	Optimum	Minimum
The server is only used for Service Apps	2 vCPU, 4GB RAM	2 vCPU, 4GB RAM
The server is only used for Service Engine	4 vCPU, 4GB RAM	2 vCPU, 4GB RAM
The server is used for Service Engine and Service Apps	4 vCPU, 8GB RAM	2 vCPU, 4GB RAM

Application Server

The requirements are given for **100 Back Office** users at peak usage so as to achieve optimal service quality.

The server is dedicated to the Service Engine application and cannot be shared.

A minimum figure is also given and represents the minimum resources to deploy, even if the number of users is far below 100.

Use case	Optimum	Minimum
The server is only used for the application server	2 vCPU, 6GB RAM	2 vCPU, 4GB RAM
The server is used for the application server and the database server	4 vCPU, 8GB RAM	4 vCPU, 8GB RAM

Database Server

The requirements are given for **100 Back Office** users at peak usage so as to achieve optimum service quality.

The resources required mainly depend on the size of your database, so that data is loaded to memory as often as possible (query optimization) and the number of CPUs is sufficient to process queries.

Once again, this figure only gives a rough idea of the target due to the many parameters involved.

The SQL Server instance is dedicated to the Service Engine application. The SQL Server can support multiple instances. However, the appropriate amount of resources (CPU, RAM, Disk space) should be allocated accordingly to the server.

	Optimum	Minimum
License for less than 100 users	4 vCPU, 16GB RAM	2 vCPU, 12GB RAM
License for more than 100 users	8 vCPU, 32GB RAM	4 vCPU, 20GB RAM

SERVER DISK SPACE REQUIREMENTS

Web Front End - Service Apps

This section suggests a configuration for the resources required to run Service Apps. It does not take into consideration additional data volumes based on your specific operating constraints such as:

- Backups performed locally prior to outsourcing.
- Storage of session files based on the project.
- Apache, PHP components, etc.
- PHP session files.

Space required:

- Service Apps kernel on each Linux server = 4GB.
- Shared (resource shared between all the web front ends for designing applications, etc.) = 20GB minimum. Can be bigger depending on the number of projects, your backups, etc. 100GB of space is usually recommended.

Web Front End - Service Engine

Mono-line: 80GB free on the web front end node.

Multi-line:

- 50GB of free disk space on each web front end node.
- Minimum 100GB of free disk space in the shared filer folder This space will vary depending on the size and number of the documents attached to incidents / changes opened by your users.

Note: If the server also plays the role of Service Apps web front end, add sufficient disk space for this type of server.

Application Server - Service Engine

80GB of free disk space.

Database Server - Service Engine

The following database groups are installed on the database server:

- Service Engine kernel databases: 10GB.
- Demo database (Config + Data): 4GB (9000 users, 40,000 pieces of equipment).
- Production database + sandbox database: Supplied empty, these databases can vary:
    - Linearly in relation to the demonstration database on the employee and equipment sections.
    - Depending on your activity for managing incidents, changes, etc. Based on our statistics, we usually recommend 1GB per 2,000 incidents / changes with an average of one attachment per request.

FRONT OFFICE (SERVICE APPS)

Terminology

Instance: Independent Apps Builder engine that will be used for version upgrades, acceptance, rollbacks.

Tenant: Secure cage on Service Apps containing applications.

Apps Connector for Service Engine: Set of files that must be placed on Service Engine web front end servers.

- They will be used for the interface between the two applications.
- They will include dedicated keys as signatures between platforms.

Trusted Identity Provider (TIP): Third-party systems such as LDAP or SSO will be used for authenticating Service Apps users.

Technical prerequisites

Service Apps Apps is deployed on the same web front ends as Service Engine, however web servers can be dedicated to this part of the solution (example: Service Apps in DMZ).

Tier	Requirements
Web server	• Type: Physical or virtual • OS: Linux • Apache 2.4.40 and above • PHP: 8.0 recommended, 7.4 supported

Securing data flows between platforms

Protection against changed URLs

When there is a data flow, Service Apps checks the authenticity of the request by using unique tokens between questions and answers.

Protection against packet capture

Data flows in both directions are signed using a pair of SSL keys (2048 bits).

Data is encrypted in AES256 using a set of private keys specific to the Service Apps platform.

Service Apps flow matrix

Source	Destination	Port	Notes
Your users	Reverse proxy Load Balancer (Optional)	443 (https)	
Reverse proxy Load Balancer (Optional)	Service Apps web server	443 (https)	
Service Apps web servers	Service Engine web servers	443 (https)	Network stream from Service Engine for querying Service Apps (EZV API)
Service Engine web servers	Service Apps web servers	443 (https)	Integration of Service Apps portals in Service Engine
Service Apps web servers	Application servers	2XXXX – 2XXXX	
Service Apps web servers	File server	445 (SMB Windows 2008/2012)	
Service Apps web servers	SMTP Server	25/465	

BACK OFFICE (SERVICE ENGINE)

Technical requirements

General overview

Tier	Requirements
Web server	• Type: Physical or virtual • OS: Linux • Apache 2.4.40 and above • PHP: 8.0 recommended, 7.4 supported
Application server	• Type: Physical or virtual • OS: Windows 2016, 2019 Server and above with the latest service pack installed. 64-bit version mandatory. SQL client: A full SQL Server client must be installed on the server in the same SQL Server version as your database and including the SQLCMD and BCP tools.
SQL Server	• Type: Physical or virtual • OS: All those supported by the database version • SQL Server: Windows SQL Server 2016 SP2, 2017, 2019 Notes: Higher versions have not yet been validated. LINUX versions are not yet supported in any version. • • Full-Text must be activated in the SQL Server instance. • The license level used must be adequate to your requirements. We recommend using at least the WEB EDITION version for a classic use, and a STANDARD version if you have more than 150 simultaneously connected users. • The license level used must be adequate for needs over 5 years. For example, the Express versions of SQL Server does not allow the use of databases larger than 10GB.

Web Front End

The installation and maintenance of the operating system as well as Apache and PHP components are your responsibility.

We provide default configuration files which comply with our recommended best practice in terms of security, resilience and maintainability. The most important technical aspects are described in the following appendices (Apache Configuration, PHP Configuration).

Application Server

The application server only runs on x64 processors.

The application server must access one of the folders published on the web server(s) or filer (Samba, etc.).

Installation and maintenance of the operating system are your responsibility.

Database Server

The installation and maintenance of the operating system as well as SQL servers are your responsibility.

Service Engine flow matrix

Source	Destination	Port	Notes
Your users	Reverse proxy Load Balancer (Optional)	443 (https)	
Reverse proxy Load Balancer (Optional)	Web servers	443 (https)	
Service Engine web servers	Application servers	2XXXX – 2XXXX	
Service Engine web servers	Service Apps web URL	443 (https)	Network stream for integrating Service Apps inService Engine
Service Engine web servers	File servers	445	
Application servers	Service Engine web servers	443 (https)	Network stream for managing outgoing REST Services. Each application server is linked to its web front end.
Application servers	LDAP server	389/636	
Application servers	POP3/IMAP server	110/995 143/993	
Application servers	SMTP server	25/465	
Application servers	SQL Server server	1433	
Service Engine web servers	Self Help web presentation servers	443 (https)	If Self Help integrated in the project

USER AUTHENTICATION

Division of roles

Authentication and authorization

For Service Engine and Service Apps, we distinguish between:

- Authentication: Confirmation of the identity of the person trying to connect.
- Authorization: What the person identified has the right to do on Service Engine and Service Apps.

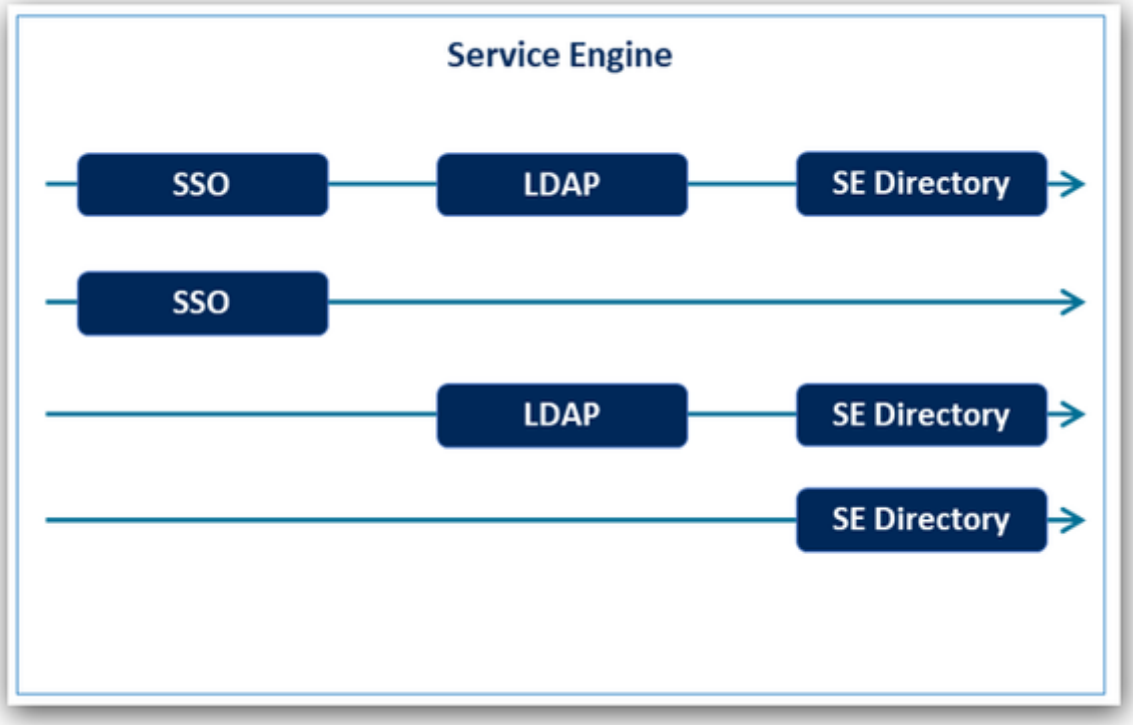
User authentication in Service Manager

Service Engine has the following means of authentication:

- Authentication via the application's internal employee database.
- Authentication via your LDAP/AD directory(ies).
- Authentication via an SSO compatible with our services.

The processing order for authorizations is as follows:

1. Identification based on SSO.
2. If step 1 is unsuccessful, authentication via login/password based on your LDAP/AD directory(ies).
3. If step 2 is unsuccessful, authentication via login/password based on the Service Engine internal directory.



Service Engine internal authentication can be deactivated. However, this deactivation is not possible if you use Service Engine as a REST service provider because, in this case, authentication is performed via the Service Engine internal directory or LDAP/AD authentication.

You can use several directories (or branches of the same directory) to authenticate your users. In this case, authentication will be performed by testing the directories in the order given.

User authentication in Service Apps

Service Apps relies on the authentication methods configured for Service Engine will be used in a transparent fashion for Service Apps.

Authorization of users

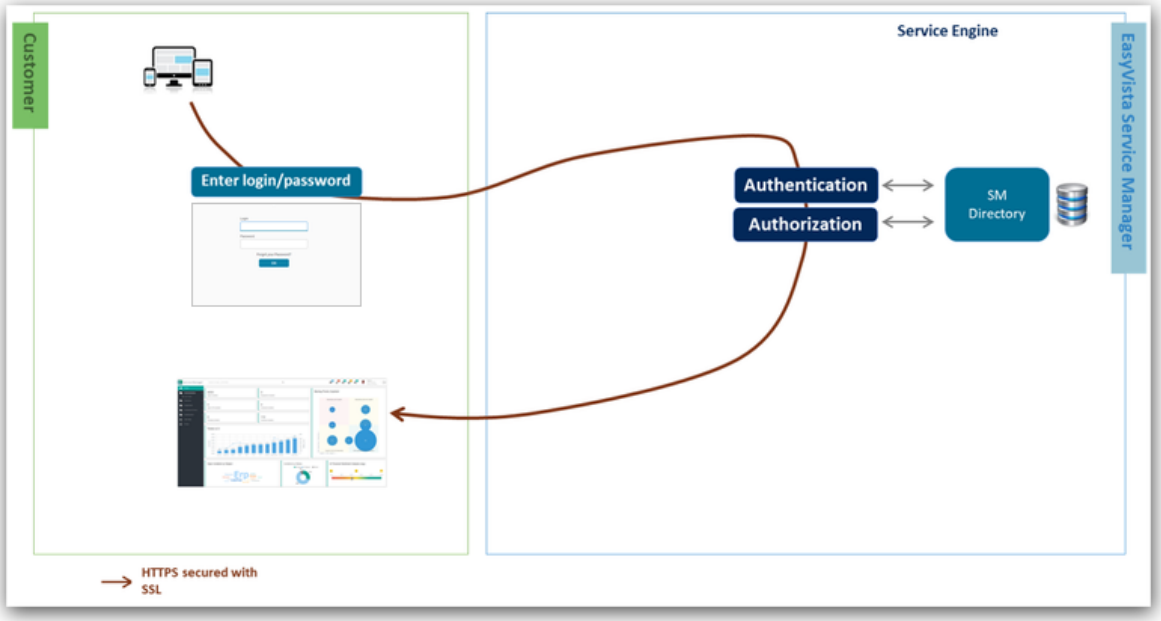
Once identified, the determination of what the user has the right to do and on what will be based on:

- Service Engine: profiles (what the user can create) and domains (what the user can see).
- Service Apps: application groups that will define the accessible applications and the role associated with the user of each of these applications.

Service Manager - Internal authentication

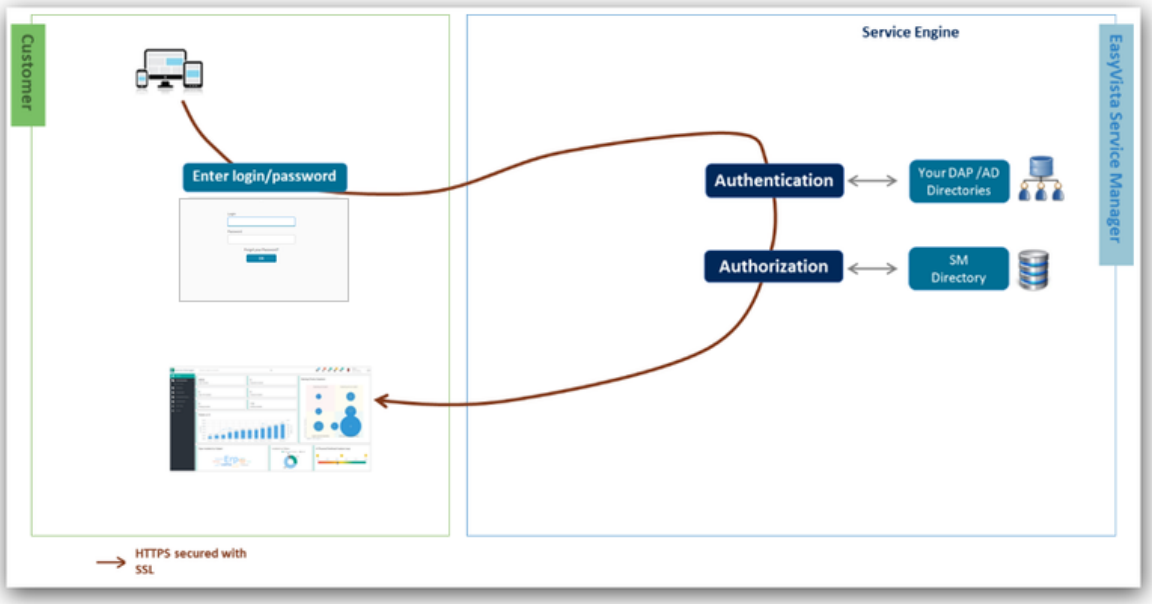
Passwords are hashed then stored (non reversible).

A policy for the length and formation can be set.



Service Manager - Authentication via LDAP/AD servers or trees

Service Engine authentication can be based on several different LDAP/AD trees.



Service Manager - SSO (Single Sign On)

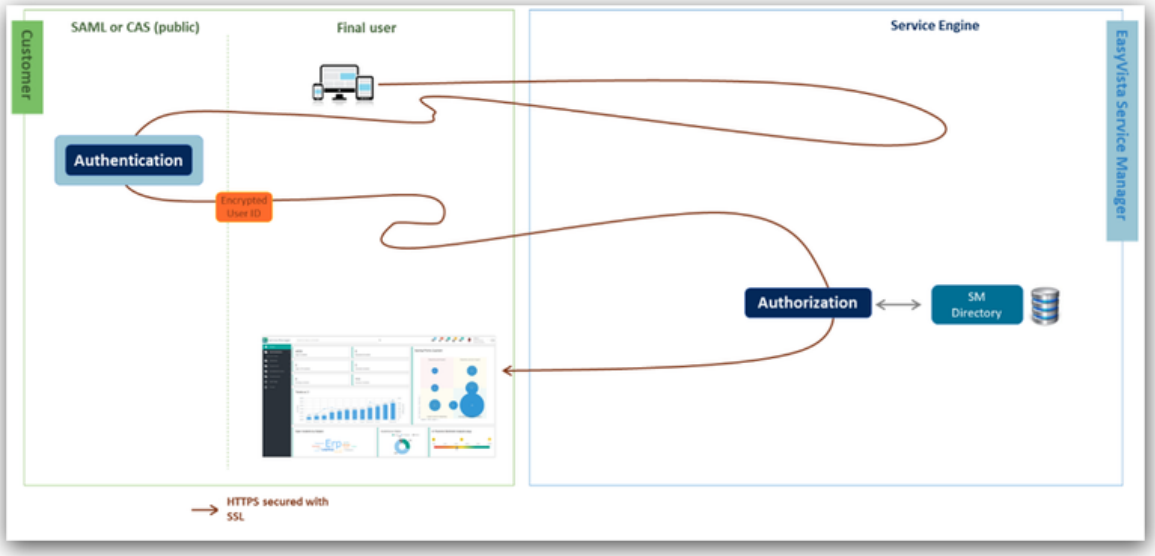
Introduction

Service Engine can authenticate users by providing identify information via our systems. The following systems are supported:

- SAML and ADFS.
- CAS.

SSO via SAML/ADFS or CAS

Your identity provider is configured in our services so that user identification is provided upon initial connection to our services.



Systems supported but not recommended

If you have an IIS server somewhere on your network, it can be used to port the identity of your user to our services.

Caution: This is not SSO, but "identity porting" (The user's identity is retrieved and transferred to our services through an encrypted header). You should consider this functionality as an easy connection for users, but in no way a completely secure solution compared to true SSO systems like SAML/ADFS or CAS which include multiple protections like:

- Derived unique key per transaction.
- Key exchange.
- Refusal of unsolicited responses.
- Impossible to perform "man in the middle" attack.
- Restriction of systems authorized to perform authentication and limited scope of accessible information.
- Traceability and alerts.

What's more, these systems rely on the fact that the user has already been identified at the network level. As a result, this will not work if our services must be accessed via a public network (which is the case for mobile applications, unless they use a VPN to simulate an internal network presence).

Identification systems specific to your company

Any identification system specific to your company can be analyzed to see how it can be used by our services (particularly in terms of availability, accessibility and security).

The analysis, development and implementation of this type of authentication will be billed separately based on their complexity.

Systems not supported or maintained

Note: This section concerns all the identification methods not previously mentioned.

While it is still technically possible to use authentication methods based on the retrieval of the identity of the user who is accessing resources (sspi, kerberos, etc.) via an Apache module, these are not supported for the following reasons (in addition to the reasons already given for IIS in the previous section):

- The Apache modules (mod sspi, mod auth kerb, etc.) have not been updated for several years and therefore have numerous security flaws.
- It is difficult to implement this type of identification with remote systems (Office 365, etc.) or different types of systems (Windows, Linux, old versions, etc.).

You will not receive assistance or maintenance if you use this type of authentication.

We recommend that you use recent SSO systems like SAML/ADFS or CAS, which guarantee both security and authentication accessibility.

AUTHORIZATION MANAGEMENT

Service Engine

In Service Engine, a user is allocated:

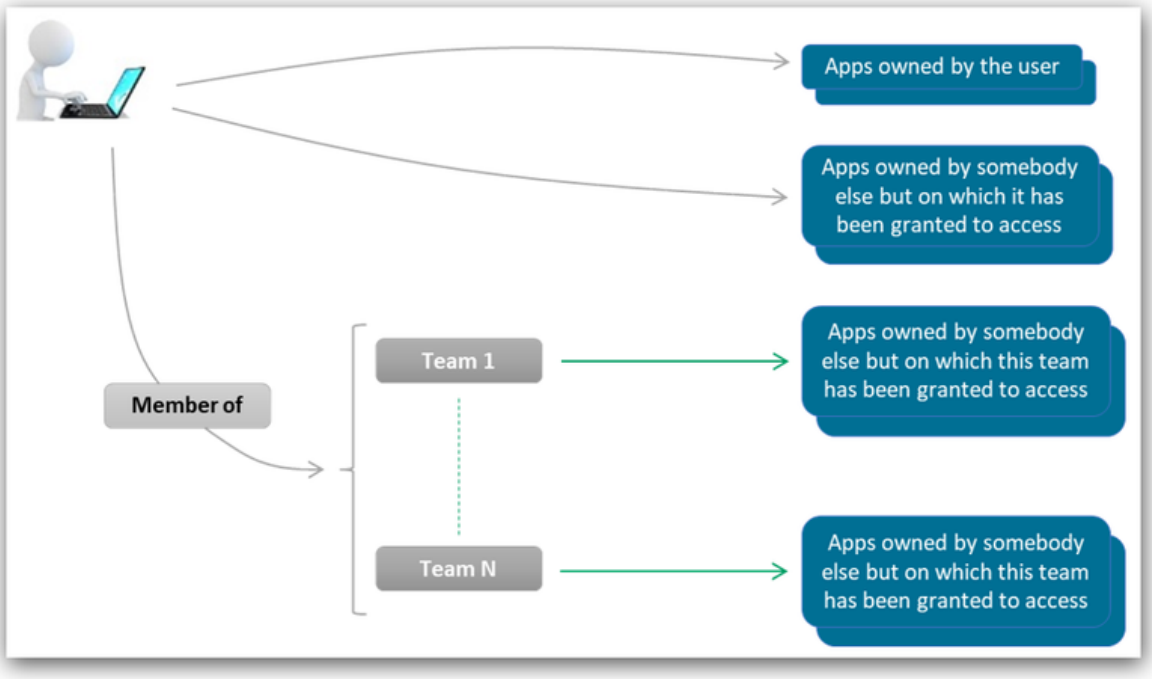
- A unique profile: This determines what the user has the right to do with all the data they have access to.
- One or more domains: These determine the scope of the date that the user has the right to access (for example, a geographical area, a type of machine with several entities, etc.).

Service Apps

How are access rights to an application granted?

A Service Apps user can be belong to several groups and have access to:

- Applications directly, because they own or have been given the right to use them.
- Applications through the teams they belong to that have the required access rights.



How are teams identified in the Service Apps directory?

In the Service Apps directory, teams are identified by their name.

These Service Apps team names are linked to the English names of Service Engine groups.

APPENDICES

Specific configuration for Windows servers

System

The "cmd" codepage must be 850 (use the "chcp" command under "cmd" to check the current status of the parameter).

Network

IPV6 is not used so you can deactivate this layer on your servers, if necessary.

The socket parameters of the IPV4 layer must be configured in the registry to handle data flows between different components.

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
MaxUserPort ->60 000
TcpTimedWaitDelay -> 30
```

They must also be configured at the NETSH level.

```
netsh int ipv4 set dynamicportrange tcp start=32767 num=32768
```

Antivirus

The local antivirus must be configured not to scan the following directories:

- Storage directory for application logs (files in XML format).
- Storage directory for SQL server databases and SQL server logs.

.NET

Version 4.5 or above must be installed on the server (depending on the configuration and version of the SQL server, a lower version may also be necessary).

Specific configuration for Linux servers

Network

IPV6 is not used so you can deactivate this layer on your servers, if necessary.

Security

SELinux in permissive mode is supported (some manual operations, like account creation before installation may, however, be necessary). In enforcing mode, an additional and specific analysis is necessary.

Specific configuration for Apache

Modules to include

- core
- so
- http
- ssl (Recommandé : 443)
- expires
- dir
- auth_basic
- access_compat
- socache_shmcb (recommandé pour cache SSL)
- reqtimeout
- filter
- deflate
- mime
- log_config
- env
- headers
- unique_id (For debug)
- setenvif
- version
- slotmem_shm
- unixd
- alias
- rewrite

- mpm_prefork (Optional: if mode=prefork)
- mpm_event (Optional: if mode=event)
- mpm_worker (Optional: if mode=worker)
- php8 or php7

If you wish to include the server-status module (optional) so you can integrate Apache monitoring into your internal monitoring tool, add the following modules:

- status
- lbmethod_byrequests
- lbmethod_bytraffic
- lbmethod_bybusyness
- lbmethod_heartbeat

To compile Apache

Use the following command as a guide, especially if you want to include socket in the compilation.

```
./configure    --prefix=/usr/local/apache2                    \
               --exec-prefix=/usr/local/apache2               \
               --sysconfdir=/usr/local/apache2/conf           \
               --with-suexec-bin=/usr/local/apache2/bin/suexec \
               --enable-authnz-fcgi                           \
               --enable-mods-shared=most                      \
               --enable-mpms-shared=all                       \
               --enable-suexec=shared                         \
               --with-apr=/usr/local/apr/bin/apr-1-config     \
               --with-apr-util=/usr/local/apr/bin/apu-1-config \
               --with-suexec-docroot=/var/www                 \
               --with-suexec-uidmin=120                       \
               --with-suexec-gidmin=120                       \
               --enable-ssl                                   \
               --enable-ssl-staticlib-deps                    \
               --with-sslport=443                             \
               --with-ssl=/usr/local/openssl                  \
               --with-mpm=prefork                             \
               --enable-static-rotatelogs                     \
               --enable-so                                    \
               --enable-info                                  \
               --enable-dir                                   \
               --enable-mime-magic                            \
               --enable-expires                               \
               --enable-headers                               \
               --enable-rewrite                               \
               --enable-cgi                                   \
               --enable-cgid                                  \
               --enable-cache                                 \
               --enable-disk-cache                            \
               --enable-mem-cache                             \
               --enable-slotmem-plain                         \
               --enable-slotmem-shm                           \
               --enable-proxy                                 \
               --enable-lbmethod-byrequests                   \
               --enable-lbmethod-bytraffic                    \
               --enable-lbmethod-bybusyness                   \
               --enable-lbmethod-heartbeat                    \
               --enable-proxy-scgi                            \
               --enable-proxy-http                            \
               --enable-proxy-ftp                             \
               --enable-proxy-fdpass                          \
               --enable-proxy-fcgi                            \
               --enable-proxy-express                         \
               --enable-proxy-connect                         \
               --enable-proxy-balancer                        \
               --enable-proxy-ajp                             \
               --enable-dav                                   \
               --enable-dav-fs                                \
               --enable-dav-lock                              \
               --enable-deflate                               \
               --with-deflate                                 \
               --with-pcre=/usr/local/pcre                    \
-- Optional: if needed to work in http2
               --with-nghttp2=/usr/local/nghttp2              \
               --enable-http2                                 \
               --enable-proxy-http2
```

Directory access security

To secure access to the directory that contains source code

```
<Directory "EasyVista_document_root">
Options -Indexes -FollowSymLinks
AllowOverride All
Order allow,deny
Allow from all
</Directory>
```

Security

```
Header edit Set-Cookie "(?i)^((?:(?!;\s?HttpOnly).)+)$" "$1; HttpOnly"
Header edit Set-Cookie "(?i)^((?:(?!;\s?Secure).)+)$" "$1; Secure"
ServerTokens Prod
ServerSignature Off
TraceEnable Off
SetEnvIfNoCase Request_URI \.(?i:gif|jpg|jpeg|pngi|jar)$ no-gzip
FileETag none
<IfModule mod_headers.c>
Header unset Server
Header unset ETag
Header set X-Frame-Options: "sameorigin"
Header append Vary User-Agent env=!dont-vary
Header set X-Content-Type-Options "nosniff"
Header set X-XSS-Protection "1; mode=block"
</IfModule>
```

Performance

```
HostnameLookups Off
Timeout 300
AddOutputFilter DEFLATE html php evsa js json htm svg gif tsv png ico css woff ttf eot

KeepAlive On
MaxKeepAliveRequests 500
KeepAliveTimeout 3
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch ".*MSIE [456].*" nokeepalive
```

Scalability

```
<IfModule prefork.c>
    StartServers           8
    MinSpareServers        8
    MaxSpareServers        30
    ServerLimit            256
    MaxClients             256
    MaxRequestsPerChild  4000
    Define PERFS_PREFORK
</IfModule>

<IfModule event.c>
    # ThreadsPerChild 10
    ThreadsPerChild 20
    ServerLimit 4
    AsyncRequestWorkerFactor 2
    # MaxRequestWorkers 40
    MaxRequestWorkers 80
    Define PERFS_EVENT
</IfModule>
```

Cache management

```
ExpiresActive On
ExpiresByType image/jpg "access plus 86400 seconds"
ExpiresByType image/jpeg "access plus 86400 seconds"
ExpiresByType image/png "access plus 86400 seconds"
ExpiresByType image/gif "access plus 86400 seconds"
ExpiresByType image/ico "access plus 86400 seconds"
ExpiresByType image/icon "access plus 86400 seconds"
ExpiresByType image/x-icon "access plus 86400 seconds"
ExpiresByType text/css "access plus 86400 seconds"
ExpiresByType text/javascript "access plus 86400 seconds"
ExpiresByType text/html "access plus 86400 seconds"
ExpiresByType application/xhtml+xml "access plus 86400 seconds"
ExpiresByType application/javascript "access plus 86400 seconds"
ExpiresByType application/x-javascript "access plus 86400 seconds"
ExpiresByType application/x-shockwave-flash "access plus 86400 seconds"
```

URL configuration

There should be direct access to the directory that contains "index.php", without using sub-levels.

Allowed: https://easyvista.mycompany.com.

Forbidden: https://projects.mycompany.com/easyista.

Log format

While this is not mandatory, complying with our format standard for access logs will simplify technical support analyses.

In certain cases, this format must be implemented, especially when the mass analysis of files is necessary. We therefore recommend that you implement this from the start of the project.

If you use SSL access:

```
LogFormat "\"%t\" \"%D\" \"%H\" \"%{Referer}i\" \"%{User-Agent}i\" \"%U\" \"%a\" \"%X\" \"%>s\" \"%b\" \"%r\" \"%{local}v:%{local}p\" \"%m\" \"%u\" \"%{X-Forwarded-For}i\" \"%{PHPSESSID}C\" \"%{email}C\" \"%{SSL_PROTOCOL}x\" \"%{SSL_CIPHER}x\" \"-\"" default_https
LogFormat "\"%t\" \"%D\" \"%H\" \"%{Referer}i\" \"%{User-Agent}i\" \"%U\" \"%a\" \"%X\" \"%>s\" \"%b\" \"%r\" \"%{local}v:%{local}p\" \"%m\" \"%u\" \"%{X-Forwarded-For}i\" \"%{PHPSESSID}C\" \"%{email}C\" \"%{SSL_PROTOCOL}x\" \"%{SSL_CIPHER}x\" \"%{Cookie}i\"" default_https_debug
SSLSessionCache shmcb:${LOG_PATH}ssl_scache.log
```

If you don't use SSL:

```
LogFormat "\"%t\" \"%D\" \"%H\" \"%{Referer}i\" \"%{User-Agent}i\" \"%U\" \"%a\" \"%X\" \"%>s\" \"%b\" \"%r\" \"%{local}v:%{local}p\" \"%m\" \"%u\" \"%{X-Forwarded-For}i\" \"%{PHPSESSID}C\" \"%{email}C\" \"-\" \"-\" \"-\"" default_http
LogFormat "\"%t\" \"%D\" \"%H\" \"%{Referer}i\" \"%{User-Agent}i\" \"%U\" \"%a\" \"%X\" \"%>s\" \"%b\" \"%r\" \"%{local}v:%{local}p\" \"%m\" \"%u\" \"%{X-Forwarded-For}i\" \"%{PHPSESSID}C\" \"%{email}C\" \"-\" \"-\" \"%{Cookie}i\"" default_http_debug
```

Specific configuration for PHP

Modules to load

- openssl
- zlib
- bcmath
- calendar
- ftp
- gettext
- mbstring
- bz2
- dba=shared
- soap
- sockets
- shmop
- exif
- intl
- unixODBC
- sysvsem
- sysvshm
- sysvmsg
- mhash
- readline
- libedit
- pdo-odbc
- zend-signals
- opcache
- curl
- apxs2
- gd
- jpeg
- png
- freetype
- zip

Note: The following options should remain enabled in PHP: Hash, Fileinfo.

Modules to load if you use SSO through SAML/ADFS or CAS

- Xml
- XmlReader
- XmlWriter

To compile PHP

You can use the following command as a guide if you wish to compile PHP on your server:

```
./configure --prefix=/usr/local/php                  \
            --with-fpm-user=www-run                  \
            --with-fpm-group=www                     \
            --with-openssl=/usr/local/openssl        \
            --with-zlib                              \
            --enable-bcmath                          \
            --enable-calendar                        \
            --enable-ftp                             \
            --with-gettext                           \
            --enable-mbstring                        \
            --with-bz2                               \
            --enable-dba=shared                      \
            --enable-soap                            \
            --enable-sockets                         \
            --enable-shmop                           \
            --enable-exif                            \
            --enable-intl                            \
            --with-unixODBC=/usr                     \
            --enable-sysvsem                         \
            --enable-sysvshm                         \
            --enable-sysvmsg                         \
            --with-mhash                             \
            --with-readline                          \
            --with-libedit                           \
            --with-pdo-odbc=unixODBC,/usr            \
            --enable-zend-signals                    \
            --enable-opcache                         \
            --with-curl=/usr/local/curl              \
            --with-apxs2=/usr/local/apache2/bin/apxs  \
            --with-gd                                \
            --with-jpeg                              \
            --with-png                               \
            --with-freetype                          \
            --with-zip                               \
```

Parameters to configure in PHP.INI

```
open_basedir must be commented out
zend_extension="/[YourFolderName]/opcache.so" short_open_tag = Off
precision = 14
zend.enable_gc = On
Expose_php = Off
error_reporting = E_ALL & ~E_NOTICE
display_errors = Off
log_errors = On
log_errors_max_len = 1024
track_errors = On
error_log = should be set
variables_order = GPCS
request_order = GP
auto_globals_jit = On
default_charset = UTF-8
file_uploads = On
default_socket_timeout = 60
max_execution_time = 300
max_input_time = 300
memory_limit = 512M
post_max_size = 800M
upload_max_filesize = 800M
max_file_uploads = 20
max_input_vars = 5000
session.save_handler = files
session.save_path = should be filled in
Session.use_cookies = On
Session.name = PHPSESSID
Session.auto_start = Off
Session.cookie_lifetime = Off
Session.serialize_handler = php
Session.gc_probability = 1
Session.gc_divisor = 1000
Session.gc_maxlifetime = 18000
Session.cache_expire = 180
Session.use_trans_sid = Off Session.hash_function = Off
Session.hash_bits_per_character = 5
```

Specific configuration for SQL Server

```
Sort order = Latin1_General_CI_AS
Mixed mode authentication required
Automatic growing of tempdb or at least 1GB
Database configured with READ_COMITTED_SNAPSHOT
FullText search must be installed and available
Max Degree of Parallelism must be 1
```