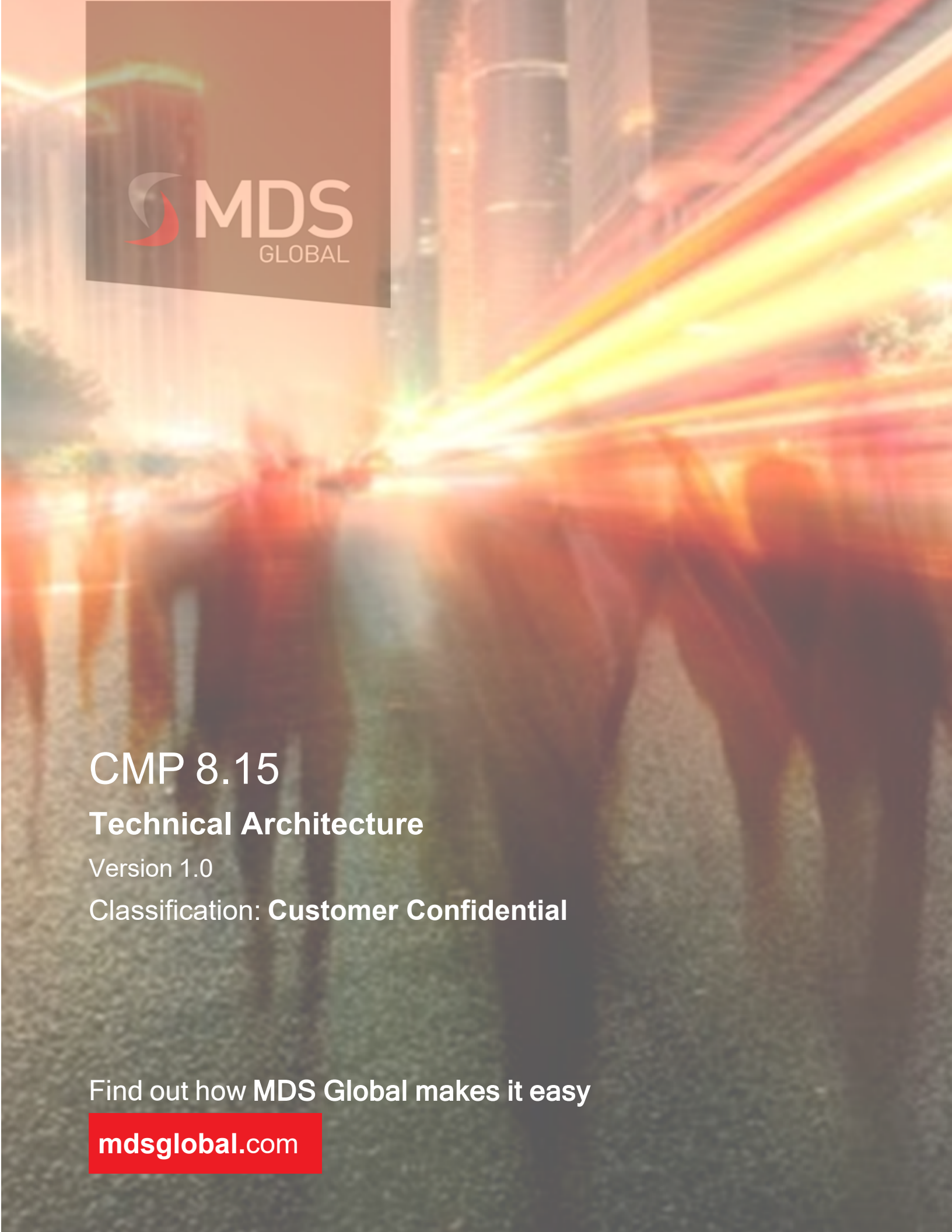


CMP 8.15

Technical Architecture

Version 1.0

Classification: **Customer Confidential**

Find out how **MDS Global makes it easy**

mdsglobal.com

Copyright

Confidentiality

Disclaimer

Trademarks

Contact Details

Please visit www.mdsglobal.com for further information on MDS Global products, solutions and services.

ISO 22301 standard is applicable to MDS Global Business Operations.

Table of Contents

Version Control

Version	Issue Date	Author	Comments
Version 1.0	21 May 2024	MDS	CMP 8.15 Release - Updated the High Availability topic to include changes to the load balancer requirements.

Terms Used in this Document

For definitions and explanations of the terms, abbreviations and acronyms used in this document, please see the *CMP Glossary* document.

1.0 CMP Architecture Overview

The Cloud Monetisation Platform (CMP) has been developed in Java to be compatible with a wide range of physical and virtual infrastructures. However, the product is certified and supported running on Red Hat Enterprise Linux or Rocky Linux, given that the installation, functionality and performance have been proven using these operating systems.

The decision to use Red Hat Enterprise Linux or Rocky Linux must be carefully considered prior to installation. Red Hat Enterprise Linux requires mandatory paid for commercial support to be subscribed for Linux, JBoss Enterprise Application Server, JBoss Web Server and AMQ. Rocky Linux is required to use WildFly, Tomcat and Artemis, all of which are community Open Source products without a commercial support option. The customer is responsible for the support of these third party products, and for the risks incurred if no commercial support is in place. Also note that these products cannot be used interchangeably, for example, Red Hat Linux with Tomcat.

CMP is made up of a number of independently installable components. An operational CMP system contains at least one instance of each component and, based on processing volumes and availability requirements, very often multiple instances of some components. Components can be installed on the same or different servers based on operational requirements. The components are described below along with the major third party software dependencies of each:

- Database

  Provides the underpinning repository for persisting CMP data supporting a central view of all CMP data.
- Identity Server

  Provides centralised user authentication and authorisation to the user across CMP including Single Sign On.
- Role Extender

  Translates business roles known to the Identity Server into more granular roles that are actually used for authorisation within CMP.
- AgentView

  The customer management interface destined for use by agents in a call centre to enrol customers and to maintain their details.
- Customer Management Web Services

  Provide a standard mechanism for online customer data interchange between CMP and other MDS Global and third party systems and applications. CMP has two complimentary sets of web services for customer data: RESTful web services and SOAP (for backward compatibility with older versions of CMP).

- Business Configuration

  Allows service providers to manage CMP business configuration via an interactive GUI.
- Configuration Web Services

  Use RESTful Web Services to manage CMP business configuration.
- AgentView Interfaces Layer

  Provides the business logic for AgentView.
- Published Interfaces Layer

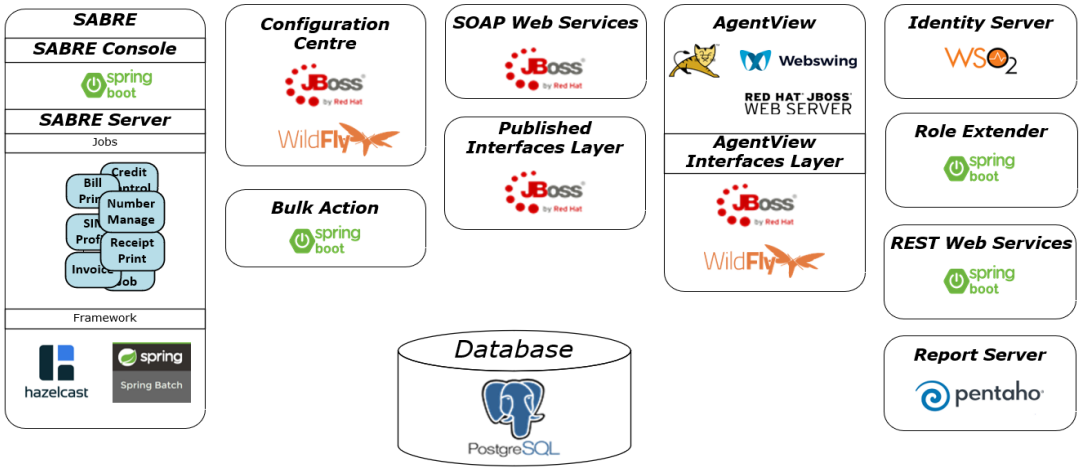
  Provides the business logic for CMP SOAP Web Services and Business Configuration.
- SABRE Server

  Executes all asynchronous processing within CMP. This includes both scheduled batch processing and immediate processing required in response to an event. The SABRE server has three parts: a technical framework, adapters, and a set of independent jobs containing processing functionality that are loaded dynamically.
- SABRE Administration Console

  The operational interface for CMP providing administration of the SABRE server. The console includes documentation of each SABRE job and allows jobs to be controlled, configured, monitored and scheduled.
- Report Server

  Allows standard CMP reports to be executed and scheduled.
- Bulk Action

  CMP Bulk Action allows users to apply bulk changes to CMP entities, for example editing subscriptions.



CMP Components Overview

1. 1 CMP Database

The CMP database holds a complete representation of the customer including all personal/business details, proposition, billing, payments and credit control data. The CMP database provides a central view of all CMP data, without the need to maintain/synchronise data across multiple databases.

The CMP database also holds the configuration used by CMP including all propositions, business process rules and reference data.

CMP does not have a strong dependency on the proprietary features of any particular relational database management system (RDBMS). The product is currently certified and tested using PostgreSQL. If required, PostgreSQL can be installed using an encrypted data partition and configured to only accept SSL encrypted network connections (automation can make this easier). PostgreSQL can support a number of failover, replication and load balancing configurations to provide scalability and high availability.

The PostgreSQL database must be monitored and managed in day-to-day operations using standard tools.

1. 2 Identity Server

The Identity Server is an instance of WSO2 and provides centralised user authentication, including Single Sign On, and role management across the different components of CMP. OpenId Connect is used for communication between the CMP components and the Identity Server. OpenId Connect focuses on authentication. It extends OAuth 2.0 to include an ID Token, which carries information about the authenticated user. This ID Token can be used by the client application to confirm the user's identity.

Identity Server Login Interface

In a standalone CMP installation, users and roles are maintained directly in the Identity Server however it can also be configured to use an external identity provider.

1. 3 Role Extender

The authorisation implementation in many parts of CMP uses very granular level roles for maximum flexibility and future proofing. It would be too cumbersome to have to grant access to all of these granular roles directly to users. A number of granular roles are therefore mapped to a higher level business roles and access is granted to these business roles.

The Role Extender, executing in Spring Boot, takes a role to which access has been granted in the Identity Server and returns the full list of lower level roles that this maps to. CMP components use roles to which that access has been directly granted and the corresponding extended lists of roles returned by the Role Extender to determine whether to allow an action to be performed.

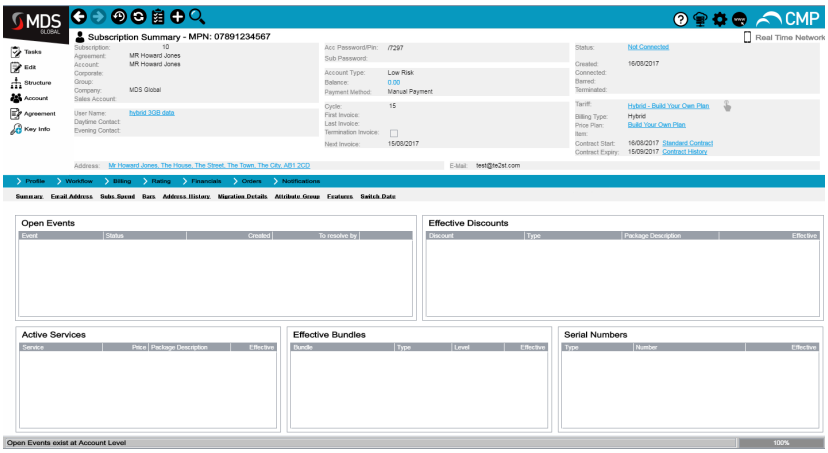
The mapping of business roles to granular roles is factory configuration that is not designed to be modified when CMP is installed.

1. 4 AgentView

AgentView is the primary user interface for CMP used by call centre staff to enrol customers onto the system and to maintain their details. The customer's personal details (name and address), proposition details (plan and tariff), and their invoice and payment preferences are all entered. Typical aspects of AgentView include:

- Create and manage customer structures and details.
- Create and connect subscriptions.
- Amend price plans.
- Manage services and discounts on a subscription.
- Answer customer queries and complaints, including access to billing and invoice information.
- Manage workflow and create tasklists for Customer Service Agents.
- Set up payment instructions and processing payments.

Many of the CMP tasks, including those above, are achieved using wizards that take the CMP user through the operation step-by-step.



AgentView User Interface

AgentView is a web client served by Webswing running in a JBoss Web Server on Red Hat Enterprise Linux or Tomcat on Rocky Linux. The AgentView application contains only presentation and validation logic. All required business logic resides in the AgentView Interfaces Layer. AgentView invokes the AgentView Interfaces Layer using RMI/JRMP. One instance of the AgentView web server connects to a single specified instance of the AgentView Interfaces Layer.

When multiple instances of the AgentView web server are installed for resilience or to manage load, connection requests should be balanced across them using a load balancer with session affinity and an extended session timeout due to the use of Web Sockets. Should an AgentView web server instance or the underlying AgentView Interfaces Layer that it is relying on fail, then all active users are required to reconnect.

Both the communication between the browser and the web server and the communication between the web server and the AgentView Interfaces can be configured to use SSL encryption if required.

1. 5 Customer Management Web Services

Customer Management Web Services allow systems and applications to create, view and manage data in the CMP database, reflecting core customer care activities that can be performed manually through the AgentView client.

CMP exposes two types of web service:

- Representational State Transfer (RESTful) web services.

  These services run in Spring Boot with in built validation, business logic and connection to the CMP database via Java Database Connectivity (JDBC).

  REST is used for all new development, and is to be used exclusively for new deployments. A full definition of available RESTful web services can be found in the *RESTful CMP API Guide*.

- Simple Object Access Protocol (SOAP) web services.

  CMP web services are exposed using SOAP over HTTP to exchange XML data.

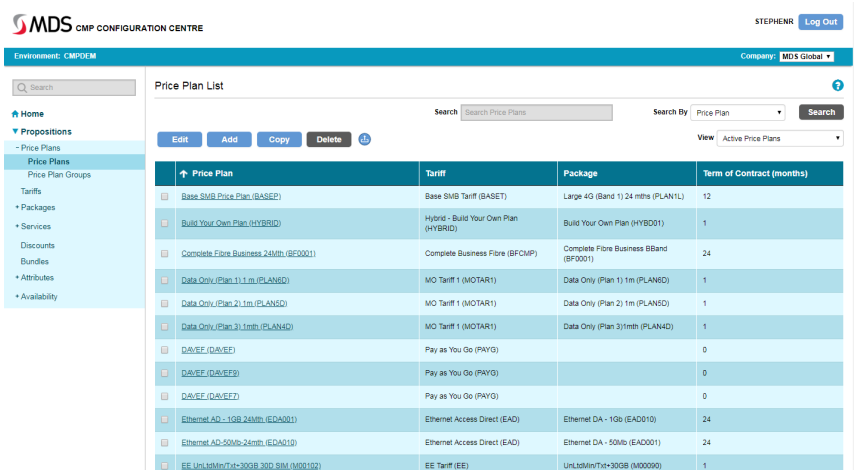
  These services run in an instance of JBoss Enterprise Application Platform (EAP) when running on Red Hat Enterprise Linux or Wildfly when running on Rocky Linux, and call the Published Interfaces Layer using RMI/JRMP for validation, business logic and communication with the CMP database.

  SOAP is used for backward compatibility with older versions of CMP.

  More technical details are provided in the *SOAP Web Services Technical Overview* and a full definition of available SOAP web services can be found in the *Web Services Semantics Guide.*

1. 6 Business Configuration

Business Configuration allows CMP business configuration to be created and maintained either via a web-based graphical interface or via a set of RESTful web services exposing the same capabilities. Service providers can therefore review and maintain their own propositions, pricing and other associated business rules either manually or programmatically. The Web Services also allow maintenance of CMP configuration to be incorporated into a wider third party configuration tool and to automate configuration deployment to multiple environments.



Business Configuration User Interface

1. 6. 1 Software Components

Business Configuration runs in an instance of JBoss Enterprise Application Platform (EAP) when running on Red Hat Enterprise Linux or Wildfly when running on Rocky Linux, and the web-based graphical interface, web services, functional business logic and connection to the database via JDBC.

The main menu of the Business Configuration console user interface includes links that allow an operator to navigate directly to the Administration Console, as described in "SABRE Administration Console" on page 12, provided access has been granted to the operator.

1.0 Configuration Web Services

Configuration Web Services enable an external system to enquire and manage the customer marketing propositions and Business Configuration data configured to support the CMP business processes. All actions that can be performed through the Business Configuration GUI are also supported through the exposed RESTful web service.

1. 7 AgentView Interfaces Layer

The AgentView Interfaces Layer contains the business logic used by the AgentView user interface to access the CMP database. The interfaces layer is accessed via RMI/JRMP and then connects to the CMP database via JDBC.

The AgentView Interfaces Layer is installed as a set of Enterprise Java Beans (EJBs) to a JBoss Enterprise Application Platform (EAP) on Red Hat Enterprise Linux, or Wildfly on Rocky Linux. The business logic is shared with the Published Interfaces Layer such that AgentView and the CMP SOAP Web Services expose functions in a consistent manner.

1. 8 Published Interfaces Layer

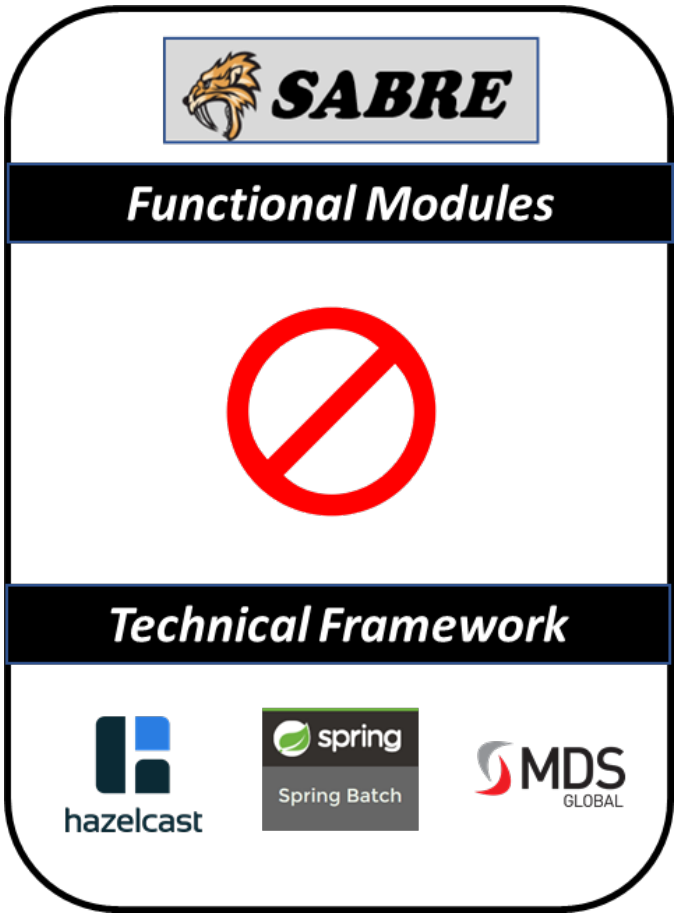
The Published Interfaces Layer contains the business logic used by the CMP SOAP Web Services. The interfaces layer is accessed via RMI/JRM and then connects to the CMP database via JDBC.

The Published Interfaces Layer is installed as a set of Enterprise Java Beans (EJBs) to a JBoss Enterprise Application Platform (EAP) on Red Hat Enterprise Linux, or Wildfly on Rocky Linux. The business logic is shared with the AgentView Interfaces Layer such that CMP SOAP Web Services and AgentView expose functions in a consistent manner.

1. 9 SABRE Server

The Secure Asynchronous Batch and Reactive Execution - or SABRE - Server provides an execution framework for all non-interactive CMP functional modules. The framework automatically handles common technical complexities such as scalability, logging, monitoring and connectivity in the background such that each functional module can be independent, as simple as possible and concentrate on addressing functional requirements.

The SABRE Server is based on Spring Batch and uses Hazelcast for in-memory storage and communication to distribute processing. Multiple instances of the SABRE Server, or nodes, can be installed in which case they form a cluster distributing processing load to provide horizontal scalability. In its simplest form, each node provides technical infrastructure but contains no functional modules.



SABRE Server Processing Architecture Overview

1. 9. 1 SABRE Functional Modules

SABRE modules are added into the SABRE server to implement functionality and, if required, can be configured to have affinity to only one or more specific nodes in the

cluster.

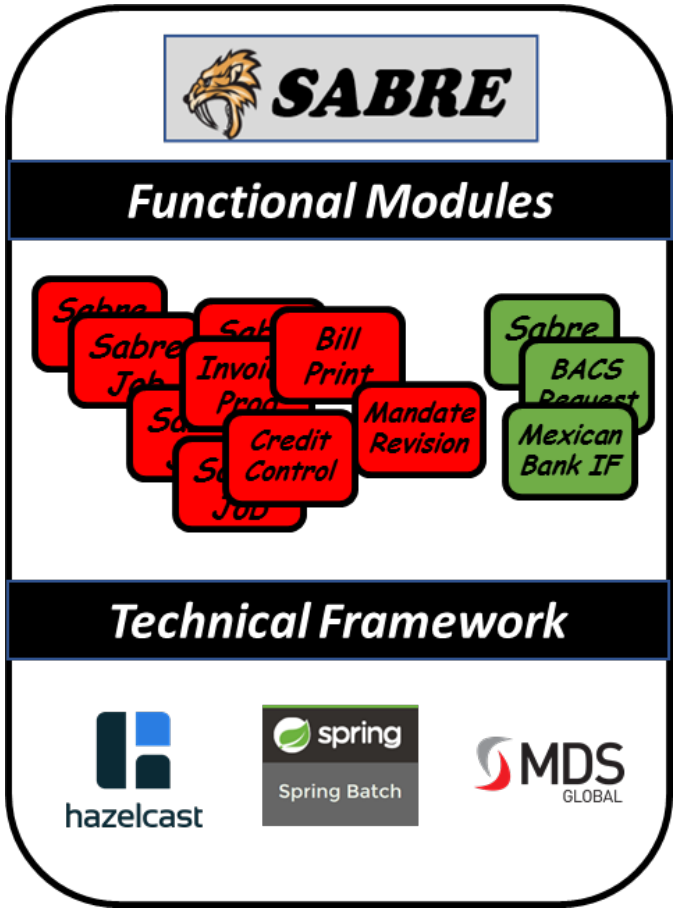
Each functional SABRE module is made up of the following elements:

- A *job* that implements the functionality and can be executed by the SABRE server based on a schedule, manually on demand or as a result of a trigger event.
- *Probes* that provide functional monitoring of a job's health. Each probe is very specific to the functionality of the job it is monitoring, for example indicating whether inputs have been received as frequently as expected or the percentage of rejections is within an acceptable tolerance. Probes can generate notifications for operators via either the SABRE console or email alerts.
- A *daemon* that performs a specific lightweight task, such as moving a file or triggering the job to execute. If required, one or more daemons are run; constantly listening for changes within the environment, for example a status change to a database record, or a file dropped to a directory.

A set of SABRE functional modules providing CMP product functionality are included in all SABRE Server installations.

All data extracted from or loaded into CMP, via a product SABRE Job, must be in the documented file format, in the majority of cases represented by a JSON schema, as described in the CMP Batch Jobs and JSON Schemas Guide.

Adapters can either be delivered as a core part of CMP, or delivered by MDS Global Professional Services as part of a customer specific project implementation. In each case, the external published interface of the relevant SABRE job is the input/output.



CMP Product and Third Party Functional Modules in a SABRE server

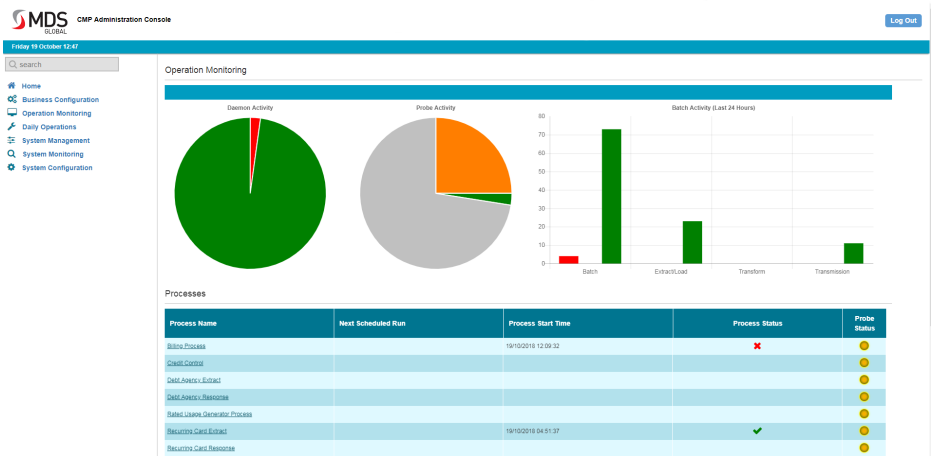
1. 10 SABRE Administration Console

The Administration Console provides an operator with an interactive environment for managing and configuring the SABRE server and the jobs, daemons and probes deployed to in functional modules.

Console users can, for example:

- Schedule jobs to run
- Run jobs on demand
- Set whether a job is to be automatically triggered
- View job, probe and daemon status
- View execution details
- View all logs in order to investigate any issues
- Display an overview of system status
- View and change technical configuration
- View the output of probes.

The SABRE Administration Console runs in Spring Boot and includes a Business Configuration link in the main menu allowing an operator to navigate directly to Configuration Centre provided access has been granted to the operator.



Administration Console User Interface

For more information on the SABRE Administration Console, see the *Online Help* provided with console.

1. 11 Report Server

The Report Server is an instance of a Pentaho Server installed with a set of standard reports ready to be run against the CMP database.

For more information about the reports installed with the Report Server, see the *Standard Reporting Guide*.

1. 12 Bulk Action

The CMP Bulk Action module runs in Spring Boot, and allows users to apply bulk changes to CMP entities, for example editing subscriptions. To perform a bulk action, users provide Excel templates that contain the data to be applied to the entities. CMP converts the Excel file into a CSV file, which is then processed by a batch job.

Users can upload templates and initiate bulk action jobs in the CMP Bulk Action console.

In the console, users can:

- View details previous bulk actions that have been performed
- Drill down into an individual bulk action's details
- Drill down into workflow error details for bulk actions associated with an asynchronous workflow

- Download reports for a bulk action
- Generate reports
- Initiate a new bulk action.

See the *Bulk Action Console online help* for more information on using the console. See CMP Batch Jobs and JSON Schemas for more information on Bulk Action jobs.

2.0 High Availability

2. 1 Introduction

As CMP will not always be considered a business-critical support system, high availability (HA) is an option as part of a deployment and not an architecture that is deployed by default. For HA, the database operates in an Active/Passive configuration while the other parts of CMP operate in an Active/Active configuration.

2. 2 Responsibilities

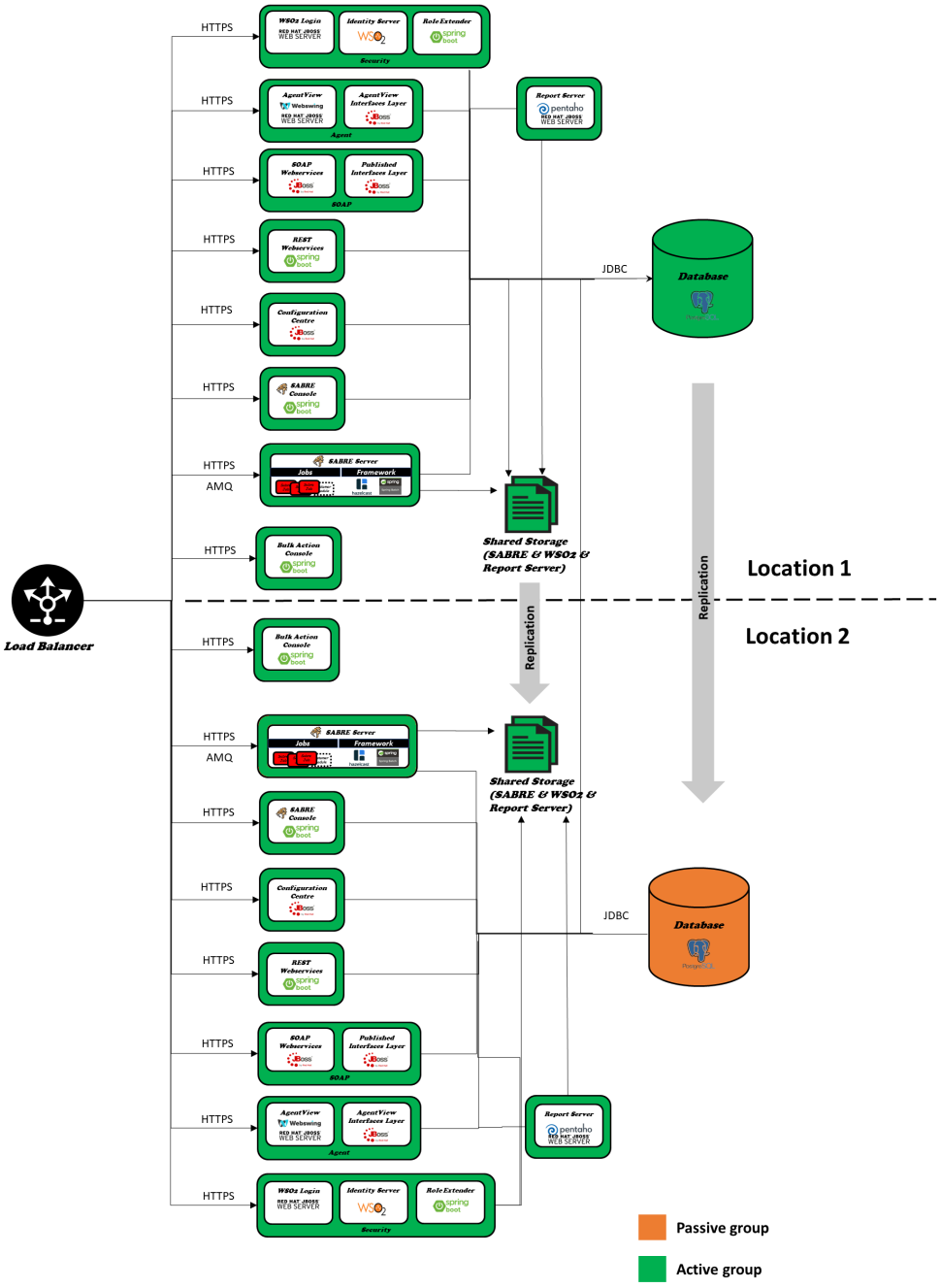
Deploying any system capable of high availability and managing it both day to day and in the case of a failure to ensure that it delivers that availability are by nature complex tasks that involve much more than simply a software application with the required capabilities.

A high availability CMP installation must therefore be planned, deployed and managed by staff with a full understanding of the principles involved and who take responsibility for the overall operating environment. This means that they need to:

- Appreciate the high availability strategy for each of the individual CMP components as described below.
- Design the physical and virtual servers, storage and network that CMP will be deployed to so that the application's high availability strategy can be effectively leveraged.
- Configure the CMP installer so as to deploy the different CMP components to leverage the application's high availability strategy.
- Install and manage load balancers to correctly distribute and direct network traffic across the different instances of the deployed CMP components as described below both in the case of normal operation, when one or more components has failed, and during an automated zero downtime upgrade.
- Configure file system and database replication as described below using the selected third party solution.
- Manage resolution of failures and, where relevant, the manual steps required to fall back to a fully highly available configuration.

2. 3 Deployment Configuration

The diagram below shows the simplest form of high availability deployment of CMP.



Active/Passive Configuration on Physical Infrastructure

All components are Active/Active except the Database. Configuring the replication and the load balancing is not part of the CMP deployment and needs to be configured using relevant third party software and hardware as a separate step to the installation itself.

The load balancer must be able to support certain functions. For more information, see High Availability Prerequisites in the *CMP Installation Guide*.

Note that the load balancer needs to be configured to expose multiple instances of the same component on a single external address. This is the case for both the external load balancer related components and the internal load balancer configured for micro services.

2. 3. 1 Zero Downtime Upgrade

The CMP installer will automatically upgrade an HA deployment with zero downtime. For this to work, the following must be in place:

- Two phases of the database upgrade (one before any software update and one after all software updates).
- The Load Balancer configured to reference the CMP Health Check service such that the upgrade process can sequentially take components offline, upgrade them and bring them back online.

Note that the automated upgrade process does not include management of Linux or PostgreSQL database major version upgrades, which are the responsibility of the customer.

3.0 Third Party Software Versions

3. 1 CMP Server Side Third Party Software

CMP uses a number of third party Open Source software components as described in "CMP Architecture Overview" on page 1. The table below shows the version of each major software component that CMP is certified against and whether the software component:

- Needs to be independently installed before deployment of CMP.
- Is deployed as a bundled part of CMP.
- Is supported by MDS Global as part of the CMP product.
- Needs to be subscribed independently if required.
- Is not supported by MDS Global as part of the CMP product, and support must be obtained independently if required.

Third Party Software	Certified Version	Installation Bundled with CMP	Supported as part of CMP Product	Potential Support Provider (if required)
Red Hat Enterprise Linux	8 (latest update)	No	No	Red Hat (mandatory)
Rocky Linux	8.x	No	No	An alternative to Red Hat Enterprise Linux
PostgreSQL	13.1	Yes - for initial software install only, not major version upgrades.	No	EnterpriseDB
JBoss Enterprise Application Platform	7.2	Yes	No	Red Hat (mandatory)
JBoss Web Server	5.0	Yes	No	Red Hat (mandatory)
WildFly	26.1.3	Yes	No	An alternative to JBoss
Spring Framework	5.3.28	Yes	Yes	Not applicable
Spring Boot	2.7.13	Yes	Yes	Not applicable
Spring Batch	4.3.8	Yes	Yes	Not applicable
Spring Security	5.7.9	Yes	Yes	Not applicable
Webswing	2.5.x	Yes	Yes	Not applicable
Hazelcast	3.12.13	Yes	Yes	Not applicable
WSO2 IS	5.11.0	Yes	No	WSO2
Pentaho	8.2	Yes	No	Hitatchi Vantara
AMQ	7.8.6	Yes	Yes	Red Hat (mandatory)
Artemis	2.19.1	Yes	No	An alternative to AMQ
Tomcat	9.0.76	Yes	No	An alternative to JWS
Ansible	2.13.x	Yes	Yes	

Third Party Software	Certified Version	Installation Bundled with CMP	Supported as part of CMP Product	Potential Support Provider (if required)
Java Development Kit (JDK)	java-1.8.0-openjdk-devel	Yes	No	Not applicable

The customer is wholly responsible for supporting the third party products mentioned above, which are not supported as part of CMP. The customer is also accountable for the management of all risks associated with using these third party products without support.

3. 2 Client Web Browser

In order to use CMP, a web browser is required to connect to the server. The web components of CMP use industry standard HTML5 features including web sockets.

Based on the advertised adherence to standards, in theory CMP is compatible with the following browsers:

- Internet Explorer 10 or later
- Edge 12 or later
- Firefox 11 or later
- Chrome 16 or later
- Safari 7 or later
- Opera 12.1 or later.

MDS Global carries out active testing against and will provide support for the following browsers:

- Chrome - latest stable version
- Microsoft Edge - latest stable version.

4.0 Third Party Licences

CMP embeds a number of third party Open Source packages as described in "CMP Architecture Overview" on page 1. The packages are provided under the terms of their respective licences as detailed below.

4. 1 Spring Framework

Provider: Spring

Website: https://spring.io/projects/spring-framework

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 2 Spring Boot

Provider: Spring

Website: https://spring.io/projects/spring-boot

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 3 Spring Batch

Provider: Spring

Website: https://spring.io/projects/spring-batch

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 4 Spring Security

Provider: Spring

Website: https://spring.io/projects/spring-security

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 5 Webswing

Provider: Webswing

Website: http://www.webswing.org

This software is provided on commercial terms and is sub licenced to CMP customers by MDS Global. Each CMP customer is licenced for a maximum 100 concurrent AgentView users in the absence of an explicit agreement to the contrary with MDS Global.

4. 6 Hazelcast Community Edition

Provider: Hazelcast

Website: https://hazelcast.com

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 7 WSO2 Identity Server Community Edition

Provider: WSO2

Website: https://wso2.com

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 8 Pentaho Server Community Edition

Provider: Hitatchi Vantara

Website: https://community.hitachivantara.com/community/products-and-solutions/pentaho/

This software is provided under the Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).

4. 9 OpenRate

Provider: Ian Sparkes

Website: https://github.com/isparkes/OpenRate

This software is provided under Apache 2.0 license agreement (http://www.apache.org/licenses/LICENSE-2.0).