



Gaming Management System 2.5  
Product Description

GTECH G2  
4th Floor  
70 Chancery Lane  
London WC2A 1AF

[www.gtechg2.com](http://www.gtechg2.com)





GTECH  
**G2GMS**  
We Speak Gaming

**Our Gaming Management System (GMS) is the power that brings it all together. GMS provides you with the ability to easily integrate today's games and the games of the future within your platform. It provides all the elements of operational infrastructure in a full end-to-end solution.**

GMS is a truly open platform that allows you to run a seamless back office across all platforms including third party game systems. It offers you a single, powerful, user-friendly interface that controls all aspects of maintenance, configuration and administration. Enabling you to fully integrate your external backoffice, gaming systems, CRM tools, payment systems and affiliate management tools into one seamless and powerful platform that gives you an unrivalled view of your player.

At GTECH G2 we believe that the consumer is the boss and GMS provides you with complete player management capabilities so that you can really understand your player's behaviour. We don't see GMS as just a database but as a tool that offers you a range of features and capabilities to not only acquire players but also increase lifetime player values.

With GMS you can be sure that under your bonnet is an engine that propels you into pole position.

**At GTECH G2 – We Speak Gaming**



**Detailed Product Information**

# 1. GMS Overview

## 1.1 Functional Overview

The Gaming Management System (GMS) is a comprehensive integration and administrative system that strengthens the gaming operator's business agility and builds the technical foundation for a successful online gaming operation. GMS provides complete control over all game systems, administration of users, partners, payment systems, and games. GMS contains all data relating to players, rewards and finances.

GMS is a one-account solution, any registered player gets a single account for all systems on all distribution channels. GMS handles the games of any game provider, whether they are developed as a part of the Boss Media gaming portfolio or by a third party game vendor.

In the following sections, an overview of the functionality, technology and organisation of GMS is presented.

## 1.2 Key Concepts

GMS gives the power to assess the operation, change it to grow and succeed in a dynamic marketplace. With GMS, it is possible to oversee all areas of the business, anticipate new consumer and technological trends and strategically adapt the business to gain on new opportunities.

### Designed to Integrate

The true strength of GMS is that it enables integration towards almost any other system on the market, whether it is a payment system, game system, affiliate system, or back office system. Complementing development work may be necessary but the sophisticated architecture keeps the effort to an absolute minimum. Naturally GMS is also designed to support integration to all Boss Media products.

### True One Account Solution for Players

GMS provides a single player account, regardless of the set up of the different game or payment system instances.

### User Friendly GMS Interface

The intuitive, user-friendly GMS interface allows easy navigation and management of online operations right from the desktop.

### 360 View for Gaming Operators and Players

GMS enables a complete 360 view for both players and GMS client users by giving full access to both future events and a complete history of expired events for the respective users. A complete overview of all ongoing events such as gaming, rewards, deposits, and withdrawals are also available for the different users.

## Reward Management

The management of rewards is as easy as can be in GMS. A bonus promotion guide helps you to set up all available rewards in GMS in an easy to use, step by step guide. Both bonus money and loyalty points are key concepts of the reward program. A player can receive rewards for events such as, registration, login, deposits, entering a promotion code, and when referring a friend. It is possible to create, view and manage rewards for each individual player or group of players.

## Multi Currency

Any currency is possible to use in GMS. The currency of the player is converted to the base currency of GMS for use in any game system. GMS handles the conversion and the external Currency Exchange Server (CES) periodically reads exchange rate data from an external supplier.

## 1.2.1 Core Responsibility

The core responsibility of GMS is to handle all information regarding players, rewards, reports and responsible gaming.

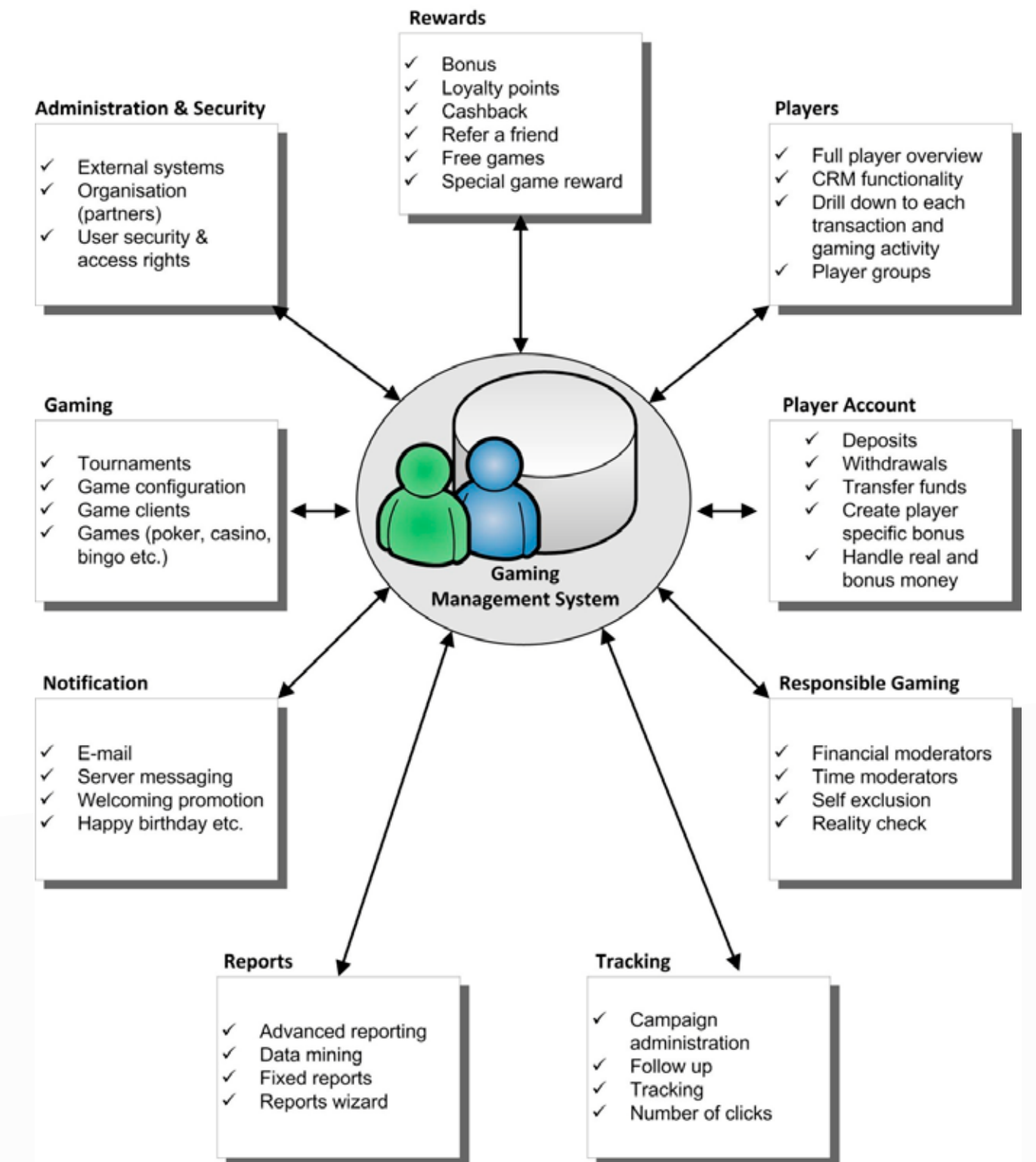


Figure 1 High level functional overview

### **1.1.3 Administration & Security**

The administration is the crossroads of GMS where the variety of systems is handled. Integration is the key to getting the big picture of your systems. Administration and addition of external systems are possible in GMS. External systems can for example be game systems, payment systems or mail servers. In GMS you can set up organisation units, such as partners, to allow easier follow up of game results and player activity. Managing partners is an important feature of GMS, which enables the GMS operator to follow up, and taking care of partners and its users.

To set up the system in a secure way, the GMS operator can set up different access rights to GMS for users or groups of users. An audit trail is also available to apply on users or functions. The audit trail lists a detailed footprint of the actions taken in GMS listed per selected user or function.

### **1.1.4 Rewards**

The GMS client user can set up a number of rewards and bonus promotions, handle payouts, conversions, and bonus money management. The main tool is the bonus promotion guide which makes it possible to create all available bonus promotions in GMS. It is possible to view promotion results and analyse future, active and expired promotions. GMS also have bonus promotion approval functionalities. This means that a reward can be set up by one user but must be approved by a user with approval permission before it becomes valid.

### **1.1.5 Players**

GMS gives you a full player overview and the possibility to drill down to each individual transaction and gaming activity. The system enables detailed management of all player finances such as transferrals of funds, deposits and withdrawals. GMS performs corrections of credited or debited money and currency exchanges in a secure way and enables you to view game replays of games that selected players have participated in.

### **1.1.6 Gaming**

In GMS you can register every single game available in your network and it is possible to monitor the games for every specific gaming portfolio. When registered, GMS collects financial data from the games, data that can be used for detailed reports or to monitor the player's progress.

### **1.1.7 Player Account**

GMS supplies easy management of the players' accounts that, for example, gives you the possibility to debit or credit the player's account for real or bonus money, handle bonuses for individuals or groups of players, lock players, add notes for players, handle password issues, view gaming results and so on.

### **1.1.8 Notification**

GMS makes it possible to increase communication with players. Notifications can be triggered by events or sent instantly on demand to both individual players and groups of players. The event notification sends e-mail messages to the player upon set conditions such as login, end game, registration, scheduled, player birthday, and so on. It is also possible to set up groups of recipients, for example according to country, tracking code or registration date. After an event notification is created, GMS administers all of this automatically without any required assistance.

### **1.1.9 Responsible gaming**

GMS offers a full suite of responsible gaming tools that helps the player to control their gaming. This is mainly done by setting limits of the gaming. Via financial and time moderators can GMS keep track of how much remains of limits that the player have set up that is left and stop the player from exceeding them. Generally these limits can be set to control the gaming per day, week and/or month.

Player may also have the need to cut them selves off from gaming temporarily or permanently, regardless of the status of any limits. The self exclusion function helps them enforce this.

With reality check players can be notified recurrently how much time that has passed and the wager and win during this specified time. All responsible gaming functionalities are configurable per partner.

### **1.1.10 Reports**

GMS contains a variety of reports, fixed and user defined. The fixed reports are the most commonly used reports and embrace players, gaming, tracking, finance and statistics. The system also features a reports wizard that makes is possible to create custom reports from the data collected by the system. You can save your reports in a personal report folder or make them available to other GMS client users. You can also schedule when and how to distribute the report. It is also possible to create reports that can be exported to external systems via web services.

### **1.1.11 Tracking**

Tracking of campaigns has never been easier, players can be grouped per tracking campaign and it is possible view specific players related to the campaign.

To add further tracking functionality to GMS, Boss Media's GMSTravis, is used to handle the collection of tracking data. Data for reports is then aggregated into the GMS database. When connected to GMSTravis, GMS provides a complete overview of number of clicks on banners, number of registered players and so on.

## 1.2 Technical Overview

This section gives an overview of the basic technology used in the Gaming Management System (GMS). For a detailed description of the product design of GMS, see chapter 4 Product Design.

### 1.2.1 High Performance and Optimal Security

GMS is built for optimal performance and security using the leading edge of technology today. The system design provides high availability at low cost of ownership. For security reasons, the system is distributed over several layers, where the top level layers access data, and functionality is accessed in the lower levels. Players will only access the top level layers, which ensures that the core data of the system will remain safely stored.

For performance reasons, the components of the system may be distributed over several physical nodes which provide performance for thousands of players on different terminals simultaneously. As the number of players increase, additional servers can easily be added. GMS uses cache functionality in the cluster to improve performance.

### 1.2.2 24 x 7 Operation

GMS is constructed for 24 x 7 operations. Stability and redundancy is a key requirement for GMS. The architecture allows the software modules to be distributed over a cluster of application servers and GMS enforces no single point of failure. All communication equipment, hardware devices and software modules can be duplicated at the game server end. For example, a network failure, corrupted disk or failed CPU or database server will not prevent GMS from continuing to operate. The network is duplicated to offer total redundancy; hence half the hardware of the system can be lost while all game clients can continue to operate. Failover functionality is provided by a load balancer.

### 1.2.3 Capacity and Scalability

The GMS architecture is designed to handle and scale to many thousands of concurrent players. This is regardless of the game type and under the assumption that the game systems check out the money during the player session. When purchases are done directly towards GMS, the load increases and the system may need to be scaled up to meet the specific requirements of the game system. It is possible to distribute load over several nodes. The scalability of the system makes it easy to scale vertically or horizontally to meet a certain performance requirement.

### 1.2.4 GMS and External Systems

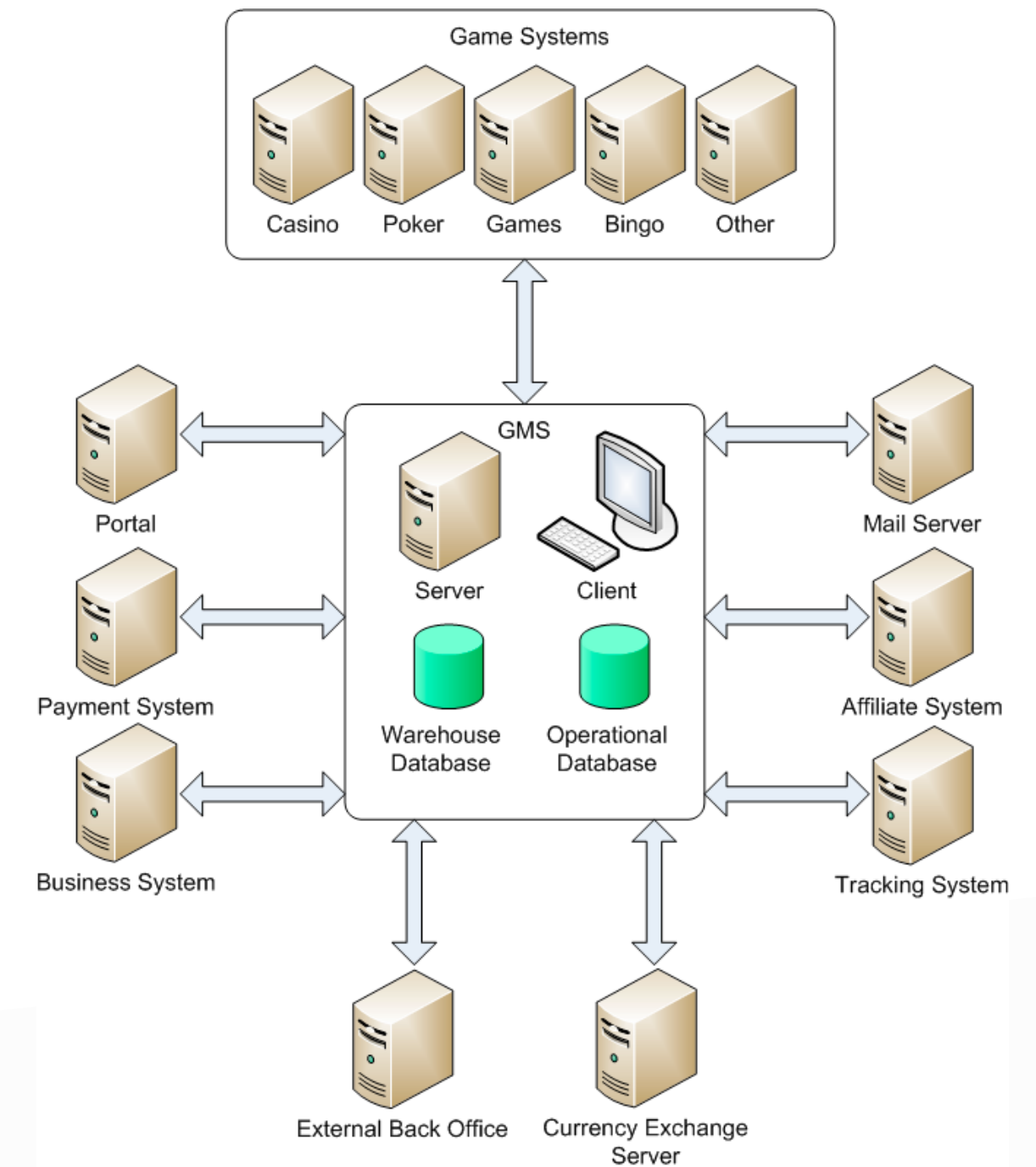
GMS consists of server, client, the operational database and the warehouse database. It aggregates information from external systems. The figure below shows an example of external systems and the relation to GMS:

Figure 2 Example overview of GMS and its related systems

GMS is built on industry standard technologies and is designed with a service oriented architecture approach to allow easy integration of other vendor systems and components or use of third-party back office tools.

### 1.2.5 Client

The GMS client is a high security and high performing application which gives the GMS operator complete control over all aspects of current gaming, gaming history, player details and more. The GMS client can interact with external applications for storage and further analysis of information gathered by GMS. The usability of the client has been developed together with Boss Media usability experts and it is regularly validated by a focus group based upon Boss Media casino and poker operators.



### 1.2.6 Server

The server system handles all logic behind GMS. This includes all functionality for external systems, all adaption of incoming data, all functionality for the GMS client, all business logic, and all read and write-operations from external sources. The server architecture is a three-layer architecture with a service layer, business logic layer and a data access layer at the bottom.

#### Service Layer

The service layer is responsible for exposing services to external systems and the GMS client. The extension API and the client services are built into the server, but the adapters are optional parts which will be included in different setups depending on the deployment scenario.

#### Extension API

The extension API exposes functionality towards external system adapters. An adapter is a component that exposes server functionality that is adapted to the needs and capabilities of one or several external systems. The extension API enables integration between GMS and external systems in a standardised way.

#### Business Logic Layer

The responsibility of the business logic layer is to handle all business logic and business rules of GMS.

#### Data Access Layer

The responsibility of the data access layer is to persist, modify, delete and read data from external and internal data sources.

### 1.2.7 External System Adapters

Adapters are responsible for adapting the protocols used by the external systems towards the extension API. An adapter is a self contained component which can be designed to use the most suitable protocol to communicate with the external system. Adapters are designed in a separate release cycle and can be redesigned without affecting the core functionality of GMS.

### 1.2.8 Database

The database of GMS is divided into two parts, the operational database and the warehouse database.

The main tasks of the operational database are to hold system configuration, player generated data from the game systems and server usage auditing logs. The operational database is configured and tuned for a fast online transaction processing function.

The main task of the warehouse database is to provide the GMS client with all report data that is needed. The warehouse database is configured for fast querying of large amounts of historical data. The warehouse database also structures this data so that it can be read in a high performance way. When structuring the data this way, the response time is kept to a minimum even if months or years of data is requested.

### 1.2.9 External Systems

External systems are systems that GMS integrates with. All systems need to be registered in GMS for the systems to be allowed to communicate with GMS. GMS presents a view of registered external systems along with belonging functionality.

## 1.3 Organisation Overview

When adding details about an organisation, a structure is created in the system which allows easy follow up on financials, results and game activities. GMS is designed to use two different organisation units:

- GMS operator – The organisation unit responsible for administrating the entire GMS installation.
- Partner – A business partner that has its own gaming licence and manages its own players with the help of the GMS client.

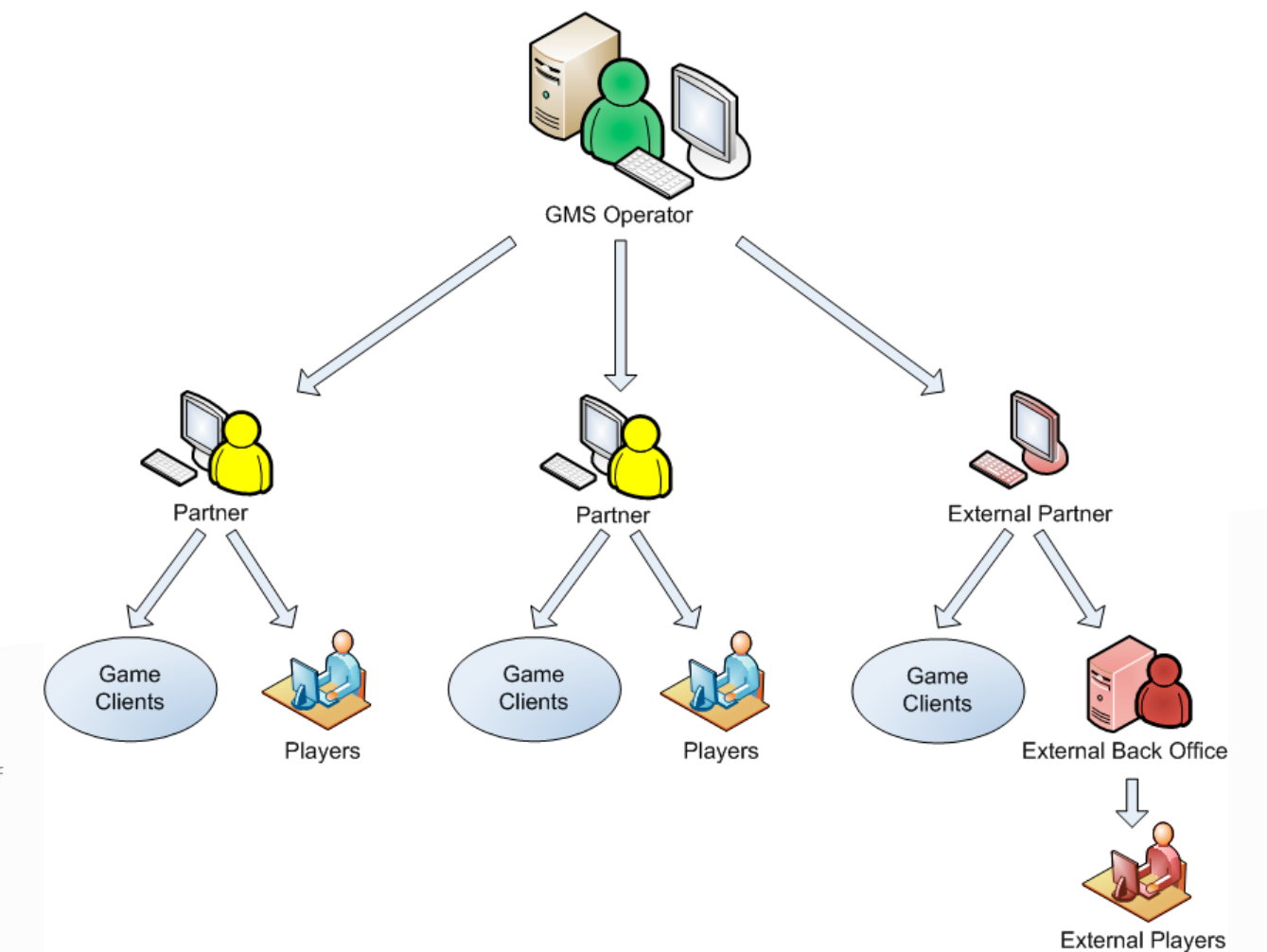
Each organisation unit in the structure can own the total configuration rights for managing all gaming and player related settings for players belonging to the respective units. For example, player messages, tracking systems, affiliate systems, reports, and bonus money management.

Figure 3 Overview of the organisation

### 1.3.1 GMS Operator

There can only be one GMS operator of an instance of GMS. The GMS operator and its users are the ones that administrate the GMS.

The GMS operator can set up any number of users and define the set of permissions for each user. The GMS operator is the only one with permission to create partners.



### 1.3.2 Partner

A partner runs the actual gaming business with the use of a GMS client while the GMS operator administrates the GMS instance. A partner can also be used as a division of the commercial operation when the GMS operator holds the gaming license.

### Players

Players are always associated with a specific partner. The players can only access game clients operated by their partner.

### 1.3.3 External Back Office

An external back office system holds the external players' master accounts. GMS monitors these accounts, without data migration, by creating a shadow account. That is, GMS mirror the external player accounts in the GMS database.

### 1.3.4 External Partner

An external partner is a partner using an external back office system. Each external partner has its own unique configuration but many external partners can share the same external back office system.

An external partner enables the creation of gaming networks where external players can participate in games played in any game systems connected to a GMS instance, as long as GMS integrates with the player's master account system.

### External Players

The term external players refer to when the player account resides in an external back office system. GMS logs the external players' progress, but it is the external back office system that is responsible for all financial transactions of external players. GMS only mirrors the external player accounts for reporting and reconciliation. The exception to this is bonus money and loyalty points management which can be handled by GMS, if so configured.

### 1.3.5 Game Client

A game client wraps one or many games from one or many game systems. A game client is always associated with a partner and it is possible for each partner to have any number of related game clients. The players belonging to a specific partner can only log on to game clients associated with that partner.

## 2 Product Functions

This chapter describes the functionality of the Gaming Management System (GMS) in relation to external systems that are possible to integrate with GMS. The chapter also describes the functionality of the GMS client.

**Note!** This section only describes the functionality of GMS, not the functionality of the external systems. For information about the product design of GMS, see chapter 4 Product Design.

Figure 4 Example overview of GMS and its related systems.

### 2.1 Client Functionality

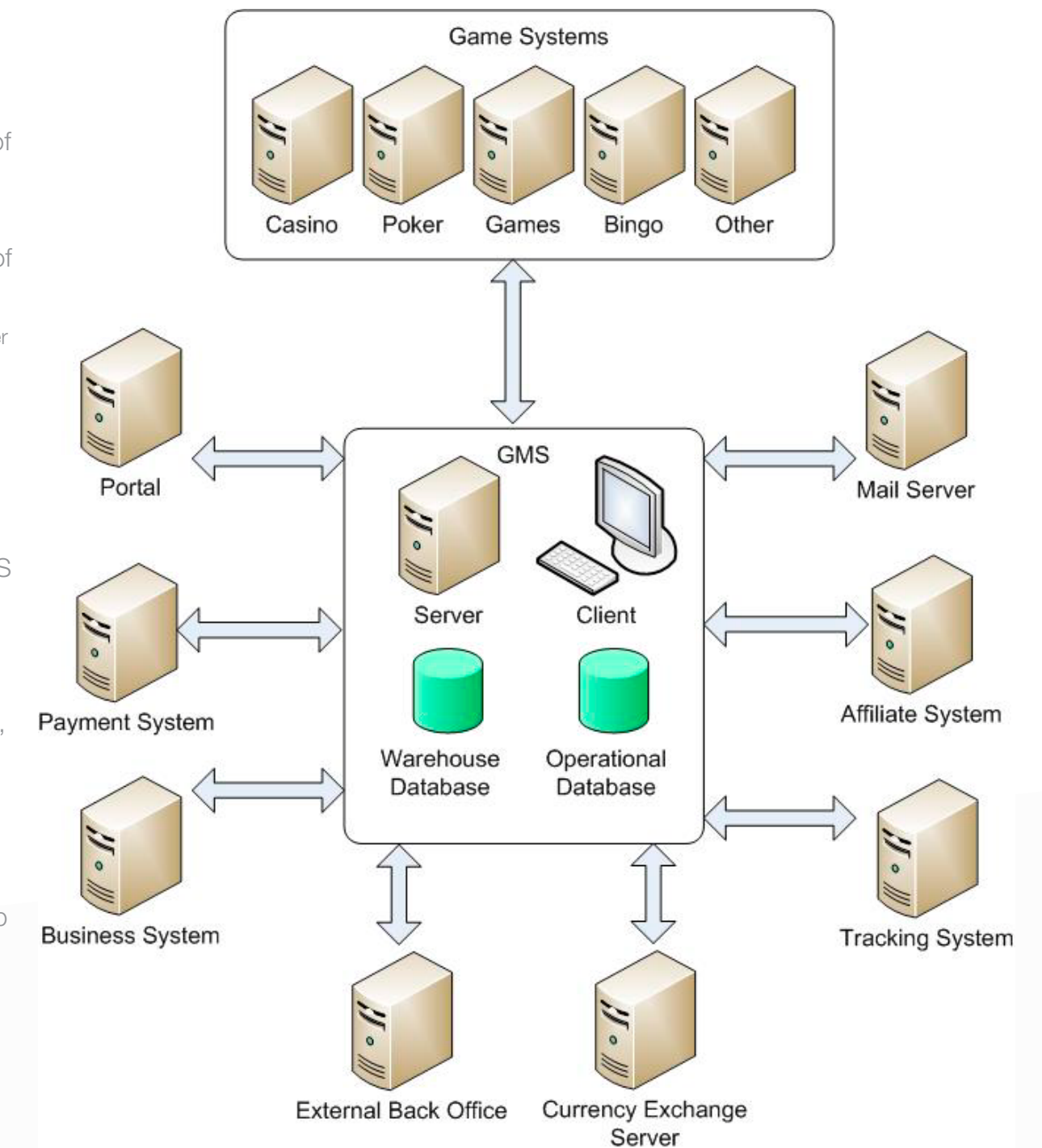
This section describes the functionalities of the GMS client from the perspective of a GMS operator.

#### 2.1.1 Administration

Administration of system settings, external systems, organisation, and security is possible in GMS with the help of the GMS client interface.

System settings include personal information for you as a GMS operator and details about the set up of the system. Available languages, countries and currencies are also available to add, edit and delete here.

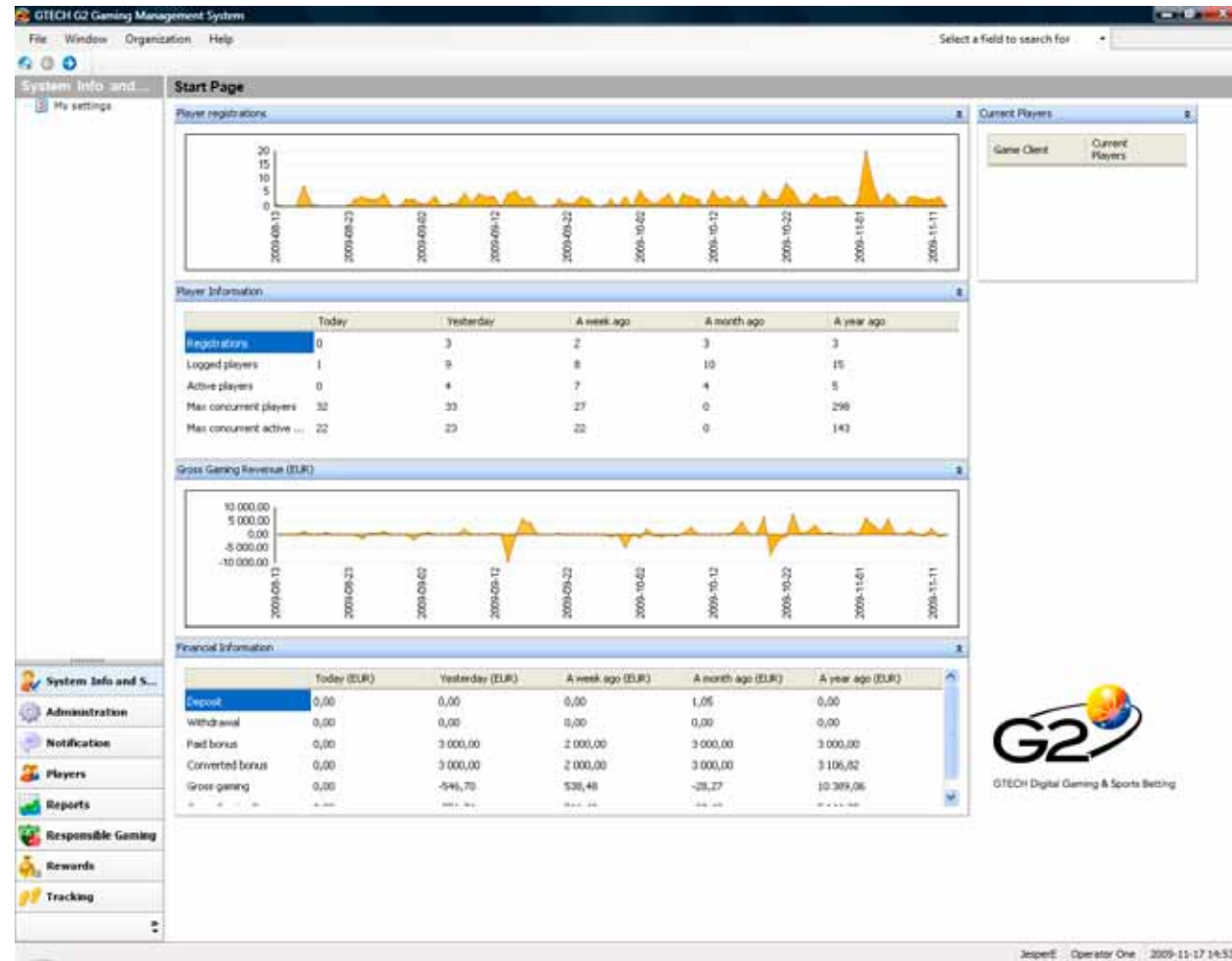
External systems contain information about all external systems connected to GMS. This includes game systems, games, game vendors, payment systems, and mail servers among other things. All external systems that are used with the support of GMS must be registered in GMS.



It is possible to add custom URLs to launch external systems from the GMS client. This functionality enables among other things to launch configuration tools for game systems. Dynamic data can be added to the URLs, sending player or game data stored in GMS to the external system.

The organisation section makes it possible to manage the organisation units of GMS and the relations between the GMS operator and the different partners or external partners. For more information about the organisation of GMS, see 2.2.9 Organisation Overview and 4.4 Organisation.

Figure 5 Example view of the GMS client



## Client Security

The GMS client provides a graphical user interface for administrating users, groups of users and permissions in the system. The functionality is logically and visually grouped into management of users and management of security groups. Each has its own set of similar forms for viewing and editing information.

### User Administration

User administration includes creating users, editing user information, locking users and assigning user permissions by adding a user to security groups or assigning specific permissions.

### Group Administration

Group administration includes creating groups, editing group information, assigning group permissions and adding users to groups.

### Partner Permission Administration

Partner permission administration allows a GMS operator to assign security actions to partners. The security actions assigned to a partner define the set of permissions that can be granted to the partner's users and groups.

## Access Groups

Access groups are groups of GMS client users with different permissions to administrate the available partner units in GMS.

## Auditing

Using the auditing function it is possible to view users currently logged on to the GMS client. A detailed footprint of each user is possible to view, this means that every action of all GMS client users are available for review.

## 2.1.2 Notification

The notification feature in GMS enables increased communication with the players. You can send notifications by e-mail that can be triggered by an event or sent on demand.

### Notification Template

You can add, edit and create e-mail templates in GMS. The WYSIWYG editor allows the GMS client user to design and maintain the templates. If a HTML-template already exists, but is not used in GMS, the source code from that template can be copied into the editor for usage in the system. The templates can then be used when you want to design customised notifications like event notification or instant notification.

### Event Notification

The event notification sends e-mails to the player when the player fulfills certain conditions like login, end game, registration, player birthday and so on. It is also possible to set up groups of recipients according to country, tracking code and gender. After an event notification is created, GMS handles the notifications automatically without any required assistance from administrators.

### Instant Notification

GMS lets the GMS client user send instant notifications by e-mail to communicate directly with players.

## 2.1.3 Players

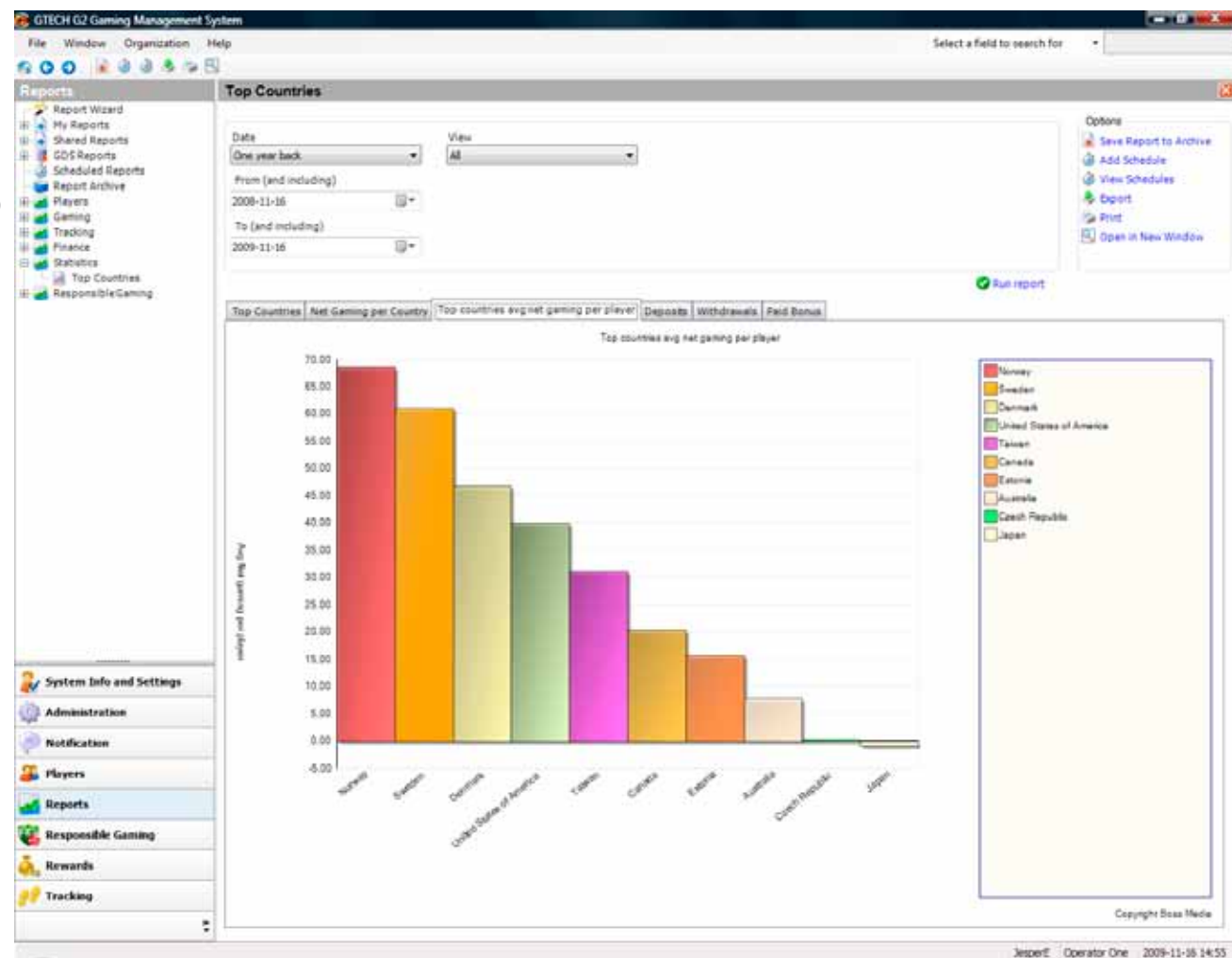
The player section contains personal information about the player such as name, address, phone number and more. It is possible to edit, add and delete this information. It is possible to change and send a new password to a player, lock the player and communicate by e-mail with the player. Login time and latest IP-address is shown in this section. Account details, player history, gaming details, transaction details, responsible gaming settings and reward details of the player are possible to view and save to file in the player section. It is also possible to statically group players according to arbitrary criteria with the player group's functionality. Player groups can then be used as a filter for promotions. Player groups in turn can again be categorized under a specific type. These group types are configurable.

## 2.1.4 Reports

GMS contains a report wizard that makes it possible to create reports from the warehouse database. There is an extensive amount of predefined variables that you can select from when creating your own report. It is possible to save the created reports and distribute them to other GMS users. In GMS, you can handle and schedule when and how to distribute the report.

Figure 6 Example of a report overview in the GMS client

In the report wizard it is possible to select which category that the report should contain and which filters to apply on the data. You can also save the report in order to make it accessible to run at any time for the person that created the report. All reports have the possibility to filter data and to order and group data by any available column.



Reports that are created can be set up for subscription by any e-mail address. The GMS client user can make a choice of how often the report should be sent (every day, a certain day of the week or a certain day of the month). It is also possible to decide what time of day the report should be sent.

You can save a created report centrally, and open it later. It is also possible to export the report to Microsoft Excel or Adobe Acrobat Reader. When exported it is possible to print the report from the respective software. The report can also be formatted as CSV (Comma Separated Values). Besides from customising your own reports, the most commonly required reports are already implemented in GMS as fixed reports.

## 2.1.5 Gaming Data Service

The Gaming Data Service (GDS) provides partners and GMS operators with the possibility to access data from the GMS warehouse database. Using the GDS to extract raw data, the partner can combine this with their own data, or view it with another tool than the GMS client. Partners can create their own reports through the report wizard to collect daily aggregated data.

### Creating the GDS Report

To be able to access data from the warehouse through GDS, a report has to be specified and saved. This is done in the GMS client with the report wizard. By setting any report created with the report wizard as a GDS report you enable the report data to be accessible by external systems through the GDS web service.

### Retrieving the GDS Report

Apart from viewing the report through GDS it is also possible to view it in the client, as with any other report created with the report wizard. The GDS makes it possible to retrieve aggregated data through SOAP and it is up to the partner to build a tool for retrieving and processing the reports.

## 2.1.6 Rewards

It is possible to create, view and manage rewards for each individual player or groups of players in GMS.

### Bonus Promotion Guide

The bonus promotion guide is a step by step guide for creating all available player bonuses in GMS. Players can receive bonuses for the following reasons (not applicable for external players):

- Registration – At registration, the player can receive a reward. This type of reward is often preceded by a marketing campaign to attract new players.
- Deposit – When depositing real money, the player can receive a reward.
- E-mail verification – When verifying the e-mail address, the player can receive a reward.
- Promotion code – the only trigger is the code in it self. External players can receive bonuses for login (only applicable for external players):
- Login bonus – External players can login to receive a reward. This is commonly aimed at inactive players to get them an incentive to start playing again. A promotion of this type is often preceded by a marketing campaign.

### Refer a Friend Program

When referring a friend, the player can receive a reward. In the refer a friend program, it is possible to set up promotions for specific gaming domains. You can choose the amount of the reward and if the reward should be paid out as bonus money or loyalty points according to your own promotion calculations. The amount of money the referred player must bet for the referring player to receive the reward is also set in the refer a friend program.

### Loyalty Points Program

In the loyalty points program you can set up different player levels according to your own loyalty points system. The amount of bonus and the requirements to receive the bonus are all available here. The loyalty points are possible to set per game or groups of games. You can also restrict the loyalty points to certain groups of players, for example, per country.

### Reward Analysis

An important part of the reward section is the possibility to analyse the result of the different promotions. With GMS you have all possibilities to get relevant and detailed information about affected players, for example, the amount of bonus money and more. All future, active and passive promotions are available for analysis. It is possible to view players and statistics related to respective campaigns and promotions results.

### 2.1.7 Tracking

It is possible to search and view information about tracking campaigns in GMS. A tracking campaign collects information on players that have clicked on banners and have been redirected to a web portal connected to GMS.

To add further tracking functionality to GMS, the Boss Media's GMSTravis, is used to handle the collection of tracking data. Data for reports is then aggregated into the GMS database. When connected to GMSTravis, GMS provides a complete overview of number of clicks on banners, number of registered players and so on.

## 2.2 Gaming

This section describes how GMS handles gaming activities in relation to a game system. A game system is an external system where the actual gaming activity takes place. Gaming activities can for example be player sessions, the life cycle of multi-player and single-player game sessions, the transferral of funds to and from player accounts and the reporting of game results.

All game systems, gaming domains, game clients, portals or games that are used with the support of GMS must be registered in GMS:

- Game system –The external system executing the game logic.
- Gaming domain – A representation of a group of games. A gaming domain can be named to match the included games or game clients.
- Game client – A game client is an entity that wraps one or many games from one or many game systems.
- Portal – An interface that the player can use to launch games.
- Game – The actual games that the player can play.

### 2.2.1 Gaming Finance

Gaming finance data is gathered from the game system and informs GMS about bets, wins and losses, which games that have been played and for how long. GMS stores financial game results in the database while all other gaming data is stored in the respective game system. However it is possible to access data stored in external systems from GMS via links to a system that for example can replay a game round.

### Transferring Funds

GMS enables money transfers to or from a game system for a player.

Three different transfer scenarios exist:

#### Transfer to Lobby

Money is transferred from GMS to a lobby and can be used for any game that can be played from the lobby. The transfer is only associated with the player session of the logged in player. The player can at any time transfer money in or out of the lobby.

#### Buy In to Game

Buy-in refers to when money is transferred from the player's account to the game at game start up. It is possible to transfer all the money automatically or to let the player enter the amount to transfer to the game. When the game session ends, the game system transfers all funds back to the player's account. The transfer is associated with a game session.

The buy-in method is recommended out of a performance perspective when running transaction intensive games, thus minimising the need of heavy money transactions between the game system and GMS (or an external back office system).

Examples of games where it would be logical to use buy in to game are transaction intensive games such as slot machines.

#### Direct Purchase

A direct purchase takes place when a game system requests money from the central player account, only enough to cover the actual bet.

The direct purchase method is often used for games which are less transaction intensive, for example, games with long game round intervals. Another use could be for games where it is more logical for the player to pay for the game directly from the central player account, without the need to first make a buy-in.

Examples of games where it would be logical to use direct purchase are: Scratch-tickets, bingo and sports betting.

### Transfer from Account

In order for a player to use money for gaming in a game system, the game system must perform a transfer from account. Transfer from account transfers money from the player's account to the game system.

A transfer from the GMS account is initiated from a game system by requesting an amount to transfer. Before any money is sent to an external system, GMS needs to verify the player's real money balance and exchange it to the currency requested by the external system. If the request and all active sessions can be verified, GMS will transfer the requested amount of money.

### Transfer to Account

In order to transfer money from a game system to a player's account, the game system must perform a transfer to account. A transfer to the GMS account is initiated from a game system. GMS exchanges the amount of money to the base currency of GMS and the currency of the player.

If the request and all active sessions can be verified, GMS will transfer the amount of money.

### Currency Exchange Rules

GMS allows players to gamble in other currencies than the currency of the player's account. The GMS operator sets a base currency, when setting up GMS, in which all monetary information must be stored. This means that when transferring money to and from GMS and calculating game result, the money is exchanged between different currencies. It is important that these exchanges are correct and lossless, that is, money should never decrease or increase in value due to exchange operations. To achieve this, GMS retrieves high precision exchange rates collected from the external Currency Exchange Server (CES) and a set of currency exchange rules. These rules are applied consequently when performing currency exchanges that affect a player's account or game result calculations.

For more information about the Currency Exchange Server, see 3.8 Currency Exchange Server.

### 2.2.2 Single Account / Single Sign-on

GMS offers single sign-on (SSO) functionality that enables the player to access many systems by a single log in. A player is authenticated in a game system by using a SSO token supplied when the player launches a new game from the portal.

Before a game can be started, using a SSO token, the player needs to log in to GMS through a portal. After the login, a SSO token request is sent to GMS and the returned SSO token is used to start a game. For every game that is started, a new SSO token needs to be acquired from GMS and be supplied with the start game call to the game system.

The game system will, before a game is started, validate the SSO token against GMS and if the token is valid, a new player session is created in GMS. A session id is returned to the game system by GMS which is later used for every call to GMS for that particular player. A new player session is then created in the game system.

Once a player is successfully logged on to a game system, the game system is responsible for handling the player session towards the game client and to authorise the user operations in the game system.

### 2.2.3 Game Sessions

GMS keeps track of player connections to external systems (for example, transfers of money or when a gaming event takes place) with the help of sessions.

#### Single-Player Game Sessions

A single-player game session represents an instance of a game played by a single participant on a game system. A single-player game session is connected to a player session and all activities regarding the game session usually requires an active player session. However, when a player joins a game in advance and the game system bets money for the player (offline purchases), no player session exists. When a game session is started, money to use for gaming can be transferred to and from the game system. Note that a game session is always associated with a certain game system and a specific game. If the game session is associated with a player session, then it is also associated with a game client via the player session.

When a game session ends, the session always returns a game result describing the outcome of the game session of the player.

There are two major ways of handling the aggregated game result and fund transfer between GMS and a game system:

- When a game session ends, the game session is closed and information about the game result and funds is transferred back to GMS. Necessary calculations using the appropriate exchange rate takes place before the information is added to the database.
- When direct purchases are used, the game result is created when the game session is initiated and kept in an open state while the game session is ongoing. During the lifetime of the game session, purchases and wins are reported to GMS and the game result is updated continuously until the game session ends. The game result is then calculated, closed and the end time is set. It is not until

#### Multiplayer Game Sessions

A multiplayer game session represents an instance of a game played by multiple participants on a game system. In order to run a multiplayer game, a game system must start a multiplayer game session in GMS.

When a multiplayer game session is initiated, individual game sessions are started for the players connected to the multiplayer game session.

The normal procedure of joining a multiplayer game session is when a player with an active game session joins.

Another way of joining a multiplayer game session is when the player makes a pre-buy. A pre-buy allows players to buy tickets for a game session in advance. After a pre-buy has taken place, it is not required of the player to be logged in when the multiplayer game session is started and played. This is known as an offline game session.

### Game Reconnect

If a disconnection occurs in the middle of an ongoing game, a reconnect is written to the game system database. The reconnect includes the unresolved pending bet of the aborted game round.

The aborted game session is terminated and the game result is sent to GMS from the game system. The pending bet is not included in the bet or jackpot contribution values, or in the closing balance. All the money in the game account except the money reserved for the pending bet is transferred back to GMS. Since the pending bet is kept in the game system, the player session can not end. When the player logs in again, the game account is re-attached to the same player session. To reconnect, a new game session is created to record the result of the restored game round and any further game rounds played immediately after restoring.

It is the game system's responsibility to send player data for reconnection. Thus, GMS never registers that a reconnect takes place, GMS only perceives an amount of money that remains in the lingering player session.

### 2.2.4 Game Result

When a game session ends, a game result, describing the outcome of the game session for the player is put together. The game result includes data such as payouts in real money amount, bonus money amount, total win amount, and so on. The game result data is stored in the database.

#### Wager

The results returned to GMS include:

- Wagered amount – The amount of real money in gaming currency, player currency and the base currency of GMS.
- Total wager amount – The total amount of real money and bonus money, in gaming currency and the base currency of GMS.

#### Win

The results returned to GMS include:

- Real Money Amount – The amount of real money in gaming currency and player currency.
- Total Win Amount – The total amount of real money and bonus money, in gaming currency and the base currency of GMS.

#### Jackpot

If the game supports jackpots, the result is collected in the game result. The game result includes data about jackpot winnings and contribution amount for the jackpot in both gaming currency and base currency.

#### Rake Results

In games where the players do not play against the house, for example poker, a small fee called rake is charged to the players at every single bet. The result of this fee is calculated within the game result, and is referred to as gross gaming revenue. Gross gaming revenue is the profit of the gaming operator.

### 2.2.5 Rewards

There are currently two types of rewards that a player can receive: Bonus money and loyalty points.

- **Bonus money** – A player can receive bonus money which can be used to bet with, but cannot be withdrawn as real money until certain conditions are fulfilled.
- **Loyalty points** – Loyalty points are accumulated and the player can exchange the points for real money.

The containers of bonus money and loyalty points hold a reward balance with the current value of the reward in the currency of the player. There are three kinds of containers for rewards:

- Bonus money bags
- Pending bonus money bags
- Pending loyalty points bags

The player can have any number of bonus bags or loyalty points bags. Each bag is connected to one or many gaming domains which restricts the usage of bonus money to games associated with that particular gaming domain. The bags can also be connected to groups of games or single games. The players can be categorised into frequent player levels of your choice, for example bronze, silver or gold. This enables the set up of bonus promotions for different groups of players.

#### Bonus Money

Bonus money is a type of reward that can be awarded to players. Bonus money is only paid out to players that have accepted to receive bonuses and that are not labelled as reward abusers. To receive bonus money, certain conditions need to be met by the player. Examples of such conditions are:

- When a certain event is triggered – Deposit, e-mail verification or other.
- When the player matches a target group – Country, tracking campaign, registration level or other.
- A time period – A time specific reward, for example, a Christmas bonus or similar.

Bonus money can have requirements attached to it that need to be met before it can be used in games. There are two kinds of requirements:

- Qualifying requirements requires the player to wager a certain amount of money before the reward becomes available for use in games. In this case the container for the reward would be a pending bonus bag.
- Wagering requirements requires the player to wager a certain amount of money before the bonus money is converted to real money. In this case the container for the reward would be a bonus money bag.

The qualifying and wagering requirement can also be combined to create a series of steps that a player need to perform before achieving the bonus.

### Transferring Bonus Money

Each game that exists in GMS is connected to a gaming domain. When a buy-in or a direct purchase scenario occurs, GMS uses the connection to the gaming domain to determine which bonus money bag that is available for the player.

When a transfer to lobby takes place, there is no game to resolve a gaming domain from. To perform a transfer of bonus money to a lobby, a game client can have a gaming domain specified. However, the specification of a gaming domain is optional, and if a game client does not specify a gaming domain, only bonus money that is not associated with a particular gaming domain is possible to transfer.

The priority of real and bonus money transactions is as follows:

- Real money is always transferred first upon a funds request from a game system.
- Bonus money is only transferred when the player's real money account balance is zero.

When a player transfers bonus money to a game system, the system must confirm that real money does not exist in the player's account in GMS or in any game systems. This keeps players from transferring real money to one game client and use the bonus money in another game client. For more information about how GMS keeps track of real and bonus money, see 3.2.6 Shadow Balance.

In a direct purchase scenario, the transferred money is consumed as soon as a purchase is completed. This allows the player an ongoing direct purchase game round with real money and at the same time use bonus money in other game systems. The use of bonus money is only possible when the player has no real money in the account or in any non-direct purchase game systems.

### Transfer Bonus Money from Account

The process is triggered when a player requests a transfer of bonus money from GMS to a game system.

- 1 GMS confirms that the player's account contains transferable bonus money for the current game system and gaming currency.
- 2 GMS performs a test that the game client can handle bonus money.
- 3 The system confirms that the player is allowed to transfer bonus money to the current game system. This is done by querying the shadow balance to check that no real money belonging to the player exists in any other game system.
- 4 If the transfer is approved, the requested amount is sent from the bonus bag to the game system.

### Transfer Bonus Money to Account

The process is triggered when a player requests a transfer of bonus money from a game system to the account in GMS.

- 1 When a game client performs a transfer to account it reports the remaining real money and bonus money balance left in the game client to enable the shadow balance to be updated with the correct data.
- 2 GMS exchanges the requested amount to the player's currency using the current player session exchange rate batch and the inverse exchange rate.
- 3 GMS collects all bonus bags in game for the player, sorted by the newest bonus bag first, and adds bonus money until the total sum is deposited. To finalise the transaction, GMS updates the shadow balance associated with the current game system.

This procedure asserts that older bonus bags will be consumed before newer bags when transferring bonus money to the player's account.

### The End Game Event

When a game ends, the reward system balances all bonus accounts, qualifying requirements and wagering requirements, performs bonus conversions and updates or deletes the current shadow balance for the game system.

That is, the end game event restores and corrects the account balance of the player and makes sure that no irregularity has taken place during the gaming event.

### Loyalty Points

A player can receive loyalty points based on a number of reasons, for example the amount wagered or when referring a friend. Loyalty points are always calculated on the end of the day results. The loyalty points can be converted to real or bonus money. External players will only receive loyalty points if their external partner has loyalty points enabled.

There are two different ways to set up loyalty points:

- The loyalty points program is an ongoing event and adds to the players pending loyalty points bag when the player wagers money.
- Loyalty points promotions are time limited campaigns for awarding extra loyalty points to a selection of players.

The players can be categorised into frequent player levels of your choice. In the loyalty points program, the frequent player level specifies deviations from the default loyalty points awarded for individual frequent player levels related to a specific game. This is done by specifying a percentage of the default value for loyalty points. For example, you can specify that players with bronze level should receive only 50% of the program points for Baccarat, while players with gold level should receive 150%.

Pending loyalty points are loyalty points that the player will receive when a qualifying requirement is met. When the qualifying requirement is met, the loyalty points will be available for conversion to real or bonus money.

For the player to qualify for a loyalty points reward, a requirement can be set based on type of game, country, player level, or a specific tracking campaign.

Player decide themselves when to convert the loyalty points. However, a lowest limit requirement can be set by the GMS client user. The lowest limit is the minimum amount of loyalty points that can be converted.

### Refer a Friend

A player can receive a reward when referring a friend to join as player. The refer a friend program can be set to apply for one or many gaming domains and all games included in the gaming domain. There are two types of reward prizes to be set, bonus money or loyalty points.

In order for any of the different rewards to be paid out, a real money bet must have been placed by the referred player. Qualifying requirements can be set to a refer a friend program. The qualifying requirement decides the amount of real money the referred player must bet in order for the referring friend to receive the reward.

### 2.2.6 Shadow Balance

Aside from the player's normal accounts, GMS also uses a shadow balance to keep track of money that a player currently has in play in a game system. A shadow balance exists for each unique ongoing game or player session.

A game session is used to keep track of the shadow balance when buy-in to game is used. Buy-in to game transfers money to a specific game, hence the shadow balance is associated to a game session.

A player session is used for keeping track of the shadow balance when money is transferred to a lobby. When a transfer to lobby takes place, there is no association with a specific game, hence the shadow balance is associated with a player session.

When real money or bonus money is transferred from GMS to a game system, GMS temporarily loses track of the credit levels within the game system. The shadow balance is used in order for GMS to keep track of the player's credit levels.

The shadow balance is the last known account balance of the player, and is continuously updated every time a game session ends or a money transfer takes place. The shadow balance enables the system to catch incorrect attempts to end the player session.

An important feature of the shadow balance is to make sure that bonus money can not be used if any amount of real money exists in the player's account or in any game systems.

### 2.2.7 Dynamic URLs

The dynamic URL feature allows a game system to request dynamic links, located in GMS, which can be used to access partner specific external services from a game system.

An example of how a request of a dynamic URL takes place:

- 1 A request of information is sent from the game system to GMS. For example when displaying player account details.
- 2 When the request is accepted by GMS, the requested information is retrieved from the GMS database.
- 3 The information is dynamically added to the URL and sent to the game system that requested the information.

Depending on the wanted behaviour of the links and the specific player logged in to the game system, a variety of information can be requested and retrieved from GMS. Dynamic links can among other things be used for log in to game systems, to launch external payment systems such as WebDollar and for displaying player account details.

The main reason for locating this functionality in GMS is that the referenced resources are typically located in systems that are connected to GMS, for example, a portal or a payment solution. The dynamic URLs normally contain information that only should be known and shared between GMS and the external systems GMS integrates to.

### 2.2.8 Game Replay

GMS enables the GMS client user to watch replays of specific games. Game replay is an external service that is called upon by GMS. The game system is responsible for the outcome of the game rounds. GMS only supplies session identifiers as a URL to the game system which presents the gaming data to the replay functionality. This enables the GMS client user to launch an external service, such as a web browser, where it is possible to watch the replay.

## 2.3 Payment

GMS handles the transfers of funds between GMS and an external payment system. The player's real money account is located in GMS. The following sections describe the payment functionality of GMS in relation to the external payment system.

### 2.3.1 Payment Transfer Types

A payment transfer type describes the type of payment used for the transfer between GMS and the external payment system. The payment can, for example, be made with VISA, Master Card or PayPal.

The type of payment is described by the external payment system and added dynamically by GMS to the GMS database if the type of payment does not already exist.

This enables any type of payment to be used even if previously unknown by GMS. The payment is performed and then added dynamically to the database as "unknown" but can later on be renamed to a correct description of the type of payment, for example, VISA.

### 2.3.2 Payment Methods

A payment method describes the type of transfer that is taking place to or from a player's real money account. The following payment methods are available:

**Deposit :** A transfer of real money from a payment system to the player's real money account located in GMS.

**Failed Deposit :** If a deposit from the payment system fails, information about this reaches GMS and a zero amount deposit is performed.

**Withdrawal :** A transfer of real money to a payment system from the player's real money account located in GMS.

**Failed Withdrawal :** If a withdrawal fails within the payment system, the money is deposited back to the player's account.

**Correction Credit :** Corrections of this kind are used by a payment system to credit a player's account in GMS. The reason for this action can vary.

**Correction Debit :** Corrections of this kind are used by a payment system to debit a player's account in GMS. The reason for this action can vary.

### 2.3.3 Discard Bonus and Pending Bonus

If a player withdraws any amount of real money before the qualifying or wagering requirements are met, the pending bonus money will be terminated. The player receives a warning message about losing the pending bonus money before the withdrawal is completed together with the option of aborting the withdrawal.

### 2.3.4 RegPixel

RegPixel is used when a third party vendor needs to know the number of first time deposits a specific campaign has generated. The idea is simple; a player creates an account in GMS after being referred from a third party campaign. The first time the player deposits real money, the payment system displays a confirmation message on a web page.

The confirmation message tells the player of the successful deposit and loads an “invisible” image on the message web page. The URL to the “invisible” image is stored in the GMS database. The image is requested via HTTP when the confirmation message loads. The image is located in the third party vendor’s external system and the third party vendor registers the requests of each image. The name of the “invisible” image indicates the tracking campaign of the player. By counting the number of requests for every specific image, the third party vendor gathers information about the number of referred players that have made a deposit.

### 2.3.5 Launch Cashier

GMS manages deposits and withdrawals by an external payment system, for example, Boss Media’s WebDollar. The URL that opens the cashier is configured in the GMS client. GMS also submits a token that the payment system can use to validate the player. When the player wants to make a deposit or a withdrawal, the cashier opens in a separate browser window. The cashier calls services in the external payment system which manages the account balance in the GMS server.

## 2.4 Player Card and Player Debit Card

GMS supports Player Card functionality. Player Card is used as identification and authorization of a player in for example lottery terminals, VLT, and IVT, i.e. used as an alternative to login with username and password.

For players using Player Cards all functionality, such as Loyalty Points and Rewards, within GMS is available just as for other players. By a third party integration the GMS also supports the use of Player Debit Card. In addition to identification and authorization this also allows players who hold player debit cards to make game related transactions as well as other routine debit card transactions, such as ATM, VISA/ MasterCard purchases etc.

All Player Card related transactions are stored and can be used for analysis of gaming and purchasing behaviour for different player segments.

## 2.5 Portal

The portal is an interface which is responsible for login functionality and enables players to launch instant games. This is handled by the sending of a token from the portal to the game system. When the player is authenticated by using a single sign-on token and permissions have been verified by GMS, a token is passed from the portal to the game.

## 2.6 Tracking

The tracking component in GMS handles tracking campaigns and other tracking related items toward an external tracking system, for example GMSTravis. The tracking component communicates directly with the tracking system through a tracking API, which means that no configuration data is stored in the GMS data source. Data for reports are stored in the warehouse database. GMS is scheduled to regularly fetch new data from the external tracking system, for example GMSTravis.

## 2.7 Affiliate

An affiliate is a business partner that promotes a gaming operator. For all players the affiliate refers to a gaming operator, the affiliate receives revenue based on the referred players’ gambling. An affiliate system provides functions for the affiliates such as creating, maintaining and following up on affiliate programs.

The affiliate functionality on the GMS server only involves exposing aggregated data for affiliate systems to calculate commission on.

### 3.8 Currency Exchange Server

The Currency Exchange Server (CES) retrieves exchange rates from an external exchange rate system. GMS collects the exchange rates from the CES. The currency functionality of GMS is responsible for managing currencies, exchange rates and supporting monetary calculations.

## 2.9 Mail Server

A mail server enables the use of e-mail communication with registered players in GMS. All mail servers need to be registered in GMS to be allowed access to the system. GMS only sends and receives mail in relation to the mail server.

## 2.10 External Back Office

The external back office (EBO) integration functionality creates generic support for integrating with third party back office systems. This enables the creation of gaming networks where external players can participate in games played in any game systems connected to a GMS instance, as long as GMS integrates with the player’s master account system.

The EBO operator gains access to all functionality GMS offers, through a GMS client, except for the management of the external player accounts which is located in the external back office system. This functionality includes reports, bonus money management and loyalty points management.

GMS only collects data when it routes information down to the external back office system. All actions on players are initiated by the external back office system while GMS performs the actions. The exception to this is the reward functionality, such as bonus money and loyalty points, where all functionality can be handled by GMS.

When an external back office system is connected, the external players' master accounts exist in the external system but are mirrored in the GMS database. The mirrored accounts, located in GMS, are referred to as shadow accounts. GMS collects the external player data and stores the information in the shadow account before routing the data to the external back office system. The shadow account exists in order for GMS to be able to manage the GMS specific functionality related to external players and to retrieve external back office data for reports in a high performance way.

### 2.11 Any Business System

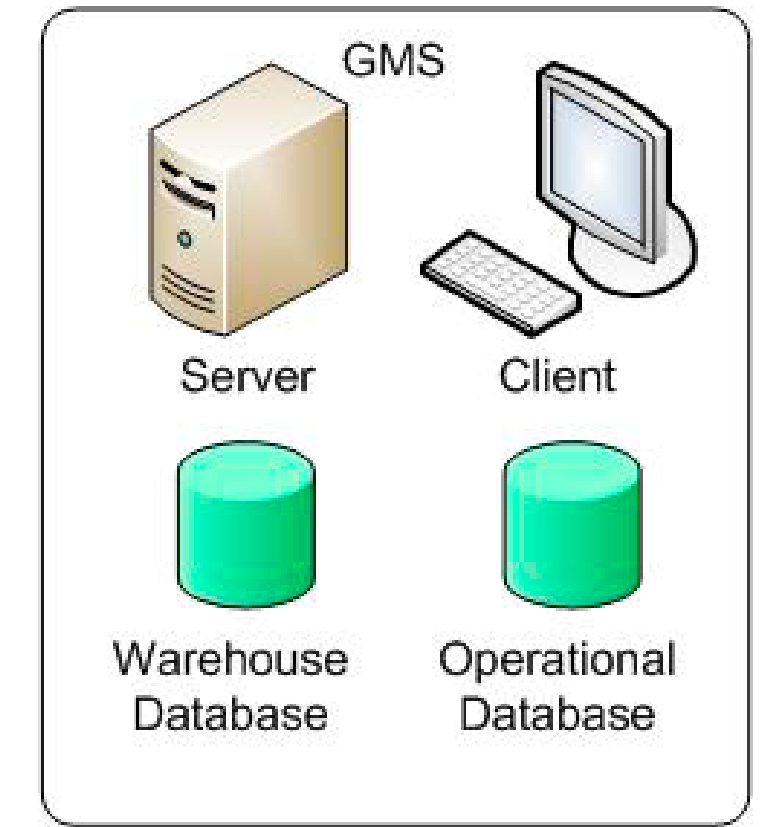
The GMS framework enables integration to almost any business system on the market. Complementing development work may be necessary but the architecture of GMS keeps the effort to a minimum.

## 3 Product Design

This chapter describes the product design of the Gaming Management System (GMS). For information about the functionality of GMS, see chapter 3 Product Functions.

GMS consists of client, server, the operational database and the warehouse database, see Figure 7 Overview of GMS.

Figure 7 Overview of GMS



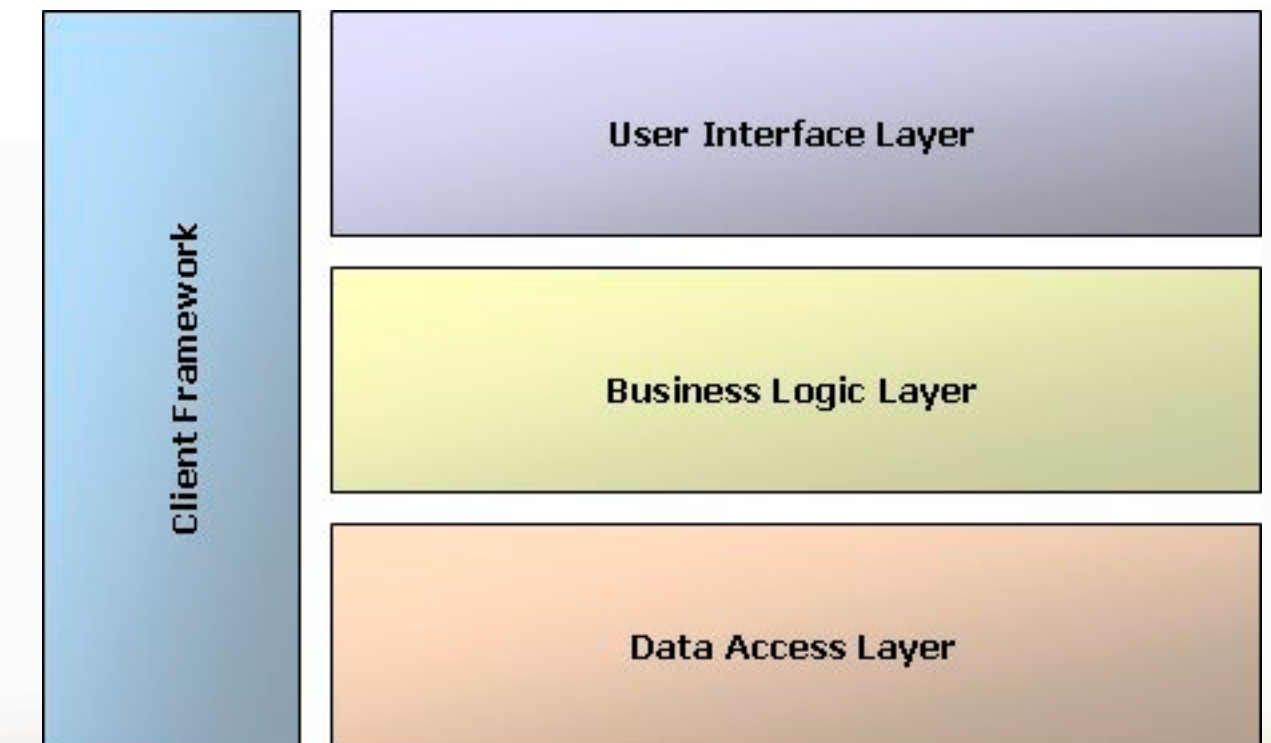
### 3.1 Client

The main business logic is located on the server, but the client uses the capabilities of the local operating system and the power of the local CPU to help relieve the pressure of the server. The data is cached locally and simple validation and state handling of the model objects are done on the client to avoid unnecessary load on the server.

The client is divided into two parts, the client framework and the client components. The client framework is the foundation of the client and acts as a runtime environment which automatically discovers and runs the components. In this way the client components are loosely linked to each other which enables addition

or removal of components and making changes to existing components without affecting other parts. All components that are designed to use a three-layer architecture, see Figure 8 Layers of a component.

Figure 8 Layers of a component



### 3.1.1 Client Framework

The client framework is designed to fulfill the following purposes:

- To provide a runtime environment for the components.
- To provide a library of reusable code that can be of benefit to the components. Examples of this are localisation, exception handling, logging, threading, GUI controls, common interfaces, utilities and base classes for all model objects.
- To make sure that the application is not dependent on a specific vendor and make it easy to change.

### 3.1.2 Design of Client Components

The client components are divided into two different categories, components and extensions. The component is a package of a generic standard functionality that handles one specific area, for example tracking, affiliate, rewards, player or notification. An extension is on the other hand a system specific add-in that extends the client in a non-generic manner.

#### User Interface Layer

- To present information to the user through a GUI.
- To interact with the user and receive response from the user.
- To initiate requests to the business logic layer to inform about events in the GUI.
- To display the interface with the correct language depending on settings.
- To only display the options and information that the user is authorised to access.

#### Business Logic Layer

The business logic layer is a preparation layer and it is responsible for the following:

- To provide the user interface layer with information and data.
- To send requests to the data access layer.
- To modify data received from the data access layer.
- To fetch locally stored data on the client computer.
- To validate user input.
- To handle errors from the data access layer.
- To transform simple arrays to business logic lists.
- Maintain instance to ensure data binding features.

#### Data Access Layer

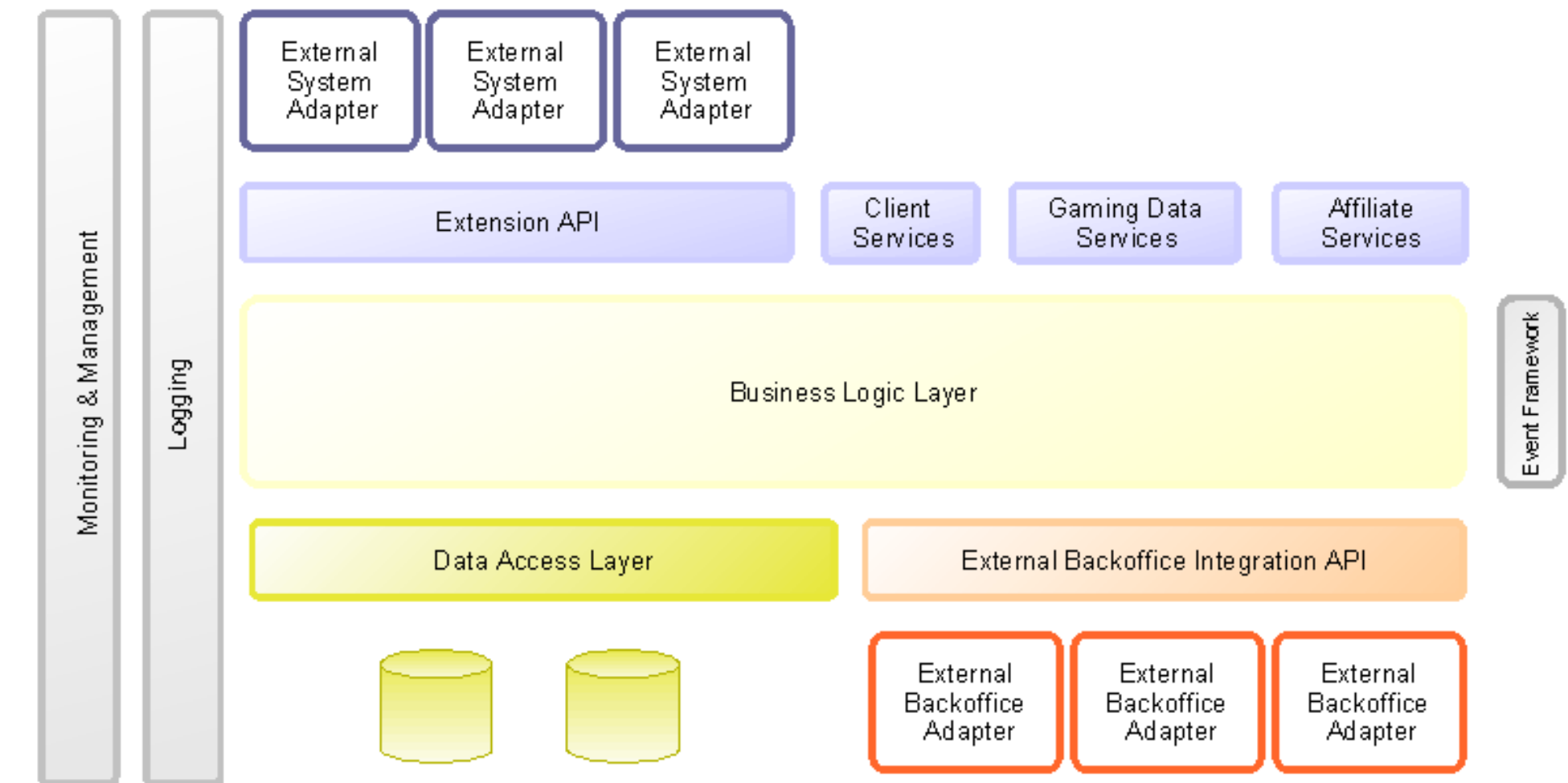
The data access layer is a communication layer that is responsible for the following:

- To communicate with the server.
- Deliver data that is received from the server to the business logic layer.
- To convert all objects to and from data transfer objects.

### 3.2 Server

The server architecture is a three-layer architecture with a service layer, business logic layer and an data access layer at the bottom. The server runs on a JBoss application server. For an overview of the different server layers, see Figure 9 Overview of server layers.

Figure 9 Overview of server layers



#### 3.2.1 External System Adapters

The adapters are responsible for adapting the protocols that the external system exposes toward the internal extension API. An adapter is a self contained J2EE component and can be configured to use the most suitable communication protocol. The existing adapters use for example serialised Java objects over HTTP, Web Services, RMI and XML over HTTP. The standard solution, and the preferred one, is however Web Services. GMS logs all communication to and from each individual adapter.

Adapters are designed in a separate release cycle and can be redesigned without affecting the core functionality of GMS. The design goals of the adapters are that they should be as thin as possible and only contain transformation logic and parts of the transaction handling depending on the external system. The external system adapters are optional parts which are used based on the deployment scenario

### 3.2.2 Service Layer

The responsibility of the service layer is to expose services to external systems and the GMS client. It is built using Java EE stateless session beans and in some cases servlets. The extension API, client services, gaming data services, and the affiliate services are built in the server.

#### Extension API

The extension API is a part of the server's service layer that exposes functionality to the external system adapters. An adapter is a component that exposes the functionality in a manner that is adapted to the needs and capabilities of one or several external systems. The extension API offers a subset of the internal functionality to enable integration between GMS and external systems in a standardised way.

All of the interfaces, model objects and exceptions that are exposed as the extension API are specific to this API. This means that the internal objects and exceptions that are returned through the API are converted to the version of the extension API. The data objects of the extension API that are sent as parameters are also converted to and from the model objects that the business logic layer uses.

The extension API is grouped based on the type of functionality it provides. These are the common functionalities: The gaming related functionality, the payment related functionality and the security functionality (for users).

#### Client Services

The client services exposes functionality towards the GMS client. The client services are built using EJB 3 stateless session beans that are exposed as web services. The web services use the Document/Literal SOAP protocol format and WSDL files are generated on installation. The only consumer of the client services is the GMS client; this means that Boss Media can handle changes to the protocol if required. The services in the client services package are grouped based on the main functionality provided by the server for the client. The following are the main packages:

- Administration – Organisation management.
- Common – Management of languages, countries, currencies and so on.
- External systems – Clients, games, game systems or organisation management.
- Notification – Administrating and sending notifications.
- Player – Player management.
- Report – Schedule and view reports (report wizard).
- Reward – Reward management and reward payouts.
- Security – User security and user session management.
- Tracking – Administration of the tracking system.

### Gaming Data Services

The Gaming Data Service (GDS) is a web service API that exposes functionality for business partners and GMS operators to access data from the GMS warehouse database.

#### Affiliate Services

The server exposes functionality to fetch data from the warehouse database relevant to an affiliate system.

The communication with the affiliate system is performed via web services. The affiliate system calls these services on a daily basis. The affiliate system also informs GMS when a new campaign is created; GMS creates the new campaign both in the external tracking system and in the warehouse database.

### 3.2.3 Business Logic Layer

The responsibility of the business logic layer is to handle all business logic and business rules. This means that the majority of the server code is located in this layer. The layer is written using Java and is wired together with Spring framework. The business logic layer is divided in several components that are packages intended to group similar functionality, for example gaming, reporting and so on.

Each component consists of a set of managers, model object and one or several data sources. A manager is responsible for managing a specific object, for example player or game session. The manager holds all the business logic for the specific object and the related activities. All managers are built using an interface that defines the behaviour of the manager and an implementation that executes the defined behaviour.

### Event Framework

The event framework holds all functionality required for messaging within GMS, but also to external systems. The event framework unifies the way events are handled in GMS, completely separating the coupling between the producer and the consumer as well as the message from the protocol. This makes it possible to implement event-driven application logic, to distribute events, perform continuous evaluations, generate statistical data, send notifications and likewise with ease.

The event framework is built upon the Spring Integration product which is Springs implementation of Enterprise Integration Patterns. It enables simple messaging, synchronous or asynchronous, within GMS and also to external systems via various protocols.

### 3.2.4 Data Access Layer

The responsibility of the data access layer is to persist, modify, delete and read data from an external data sources. All data sources are hidden behind interfaces and the implementation is located in sub packages inside each component and each component has its own set of data source. The data access layer is built using Java objects and Hibernate for O/R mapping.

### 3.2.5 External Back Office Integration API

The external back office integration API exposes functionality to the external back office adapters.

### 3.2.6 External Back Office Adapters

The adapters are responsible for adapting the protocols that the external back office system exposes toward the external back office integration API and are deployed dynamically into the server.

The client part of an external back office adapter can utilise whichever communication protocol that the solution requires, where web services are the standard.

### 3.2.7 Monitoring and Management

The GMS server contains built-in functionality that can be used for monitoring or troubleshooting live test or production environments. Most of the monitoring functionality is always available and requires no prior configuration, although some of the features must be manually activated after server start up.

The monitoring and management of the server is done with the help of Springs support for exposing Java beans as MBeans for usage via JMX.

### 3.2.8 Logging

Logging is done via the log4j framework. Configuration of logging is done in the log4j configuration file in the JBoss deploy directory.

## 3.3 Database

The database of GMS is divided into two parts, the operational database and the warehouse database.

### 3.3.1 Operational Database

The main tasks of the operational database are to handle system configuration (via the GMS client), storage of player generated data from the game systems and to log server usage for auditing. The operational database is configured and tuned for a fast online transaction processing function.

The operational database consists of the following different database tables:

- Audit – Stores an audit trail of database changes that are made by back office users, players and external systems in GMS.
- Core – The central tables in the database that are commonly used throughout the system. These tables are external systems, job scheduling, organisation and regional.
- Gaming – Stores configuration of gaming entities (systems, clients and games), transfers of player funds and other gaming events (player sessions and game rounds).
- Notification – Stores messaging configuration that the server triggers when sending out messages to players.
- Payment – Keeps track of all player deposits and withdrawals of real money funds that are performed towards an external payment system.
- Player – This part of the database stores information about the players.

- Reporting – This part stores report configurations, permissions and runs scheduling.
- Rewards – Stores promotion configuration that causes a certain player action (for example registration or deposit) to trigger a promotion evaluation and a possible reward payout.
- Security – Stores the user authorisation information, session permissions for the GMS client.

### 3.3.2 Warehouse Database

The main task of the warehouse database is to provide the GMS client with all report data that is needed. The warehouse database is configured for fast querying of large amounts of historical data. The warehouse database is used to extract raw data from the data source systems, at the moment these are the operational database and the external tracking system, for example GMSTravis. The warehouse database also structures this data so that it can be read in a high performance way. When structuring the data this way, the response time is kept to a minimum although months or years of data is requested.

## 3.4 Organisation

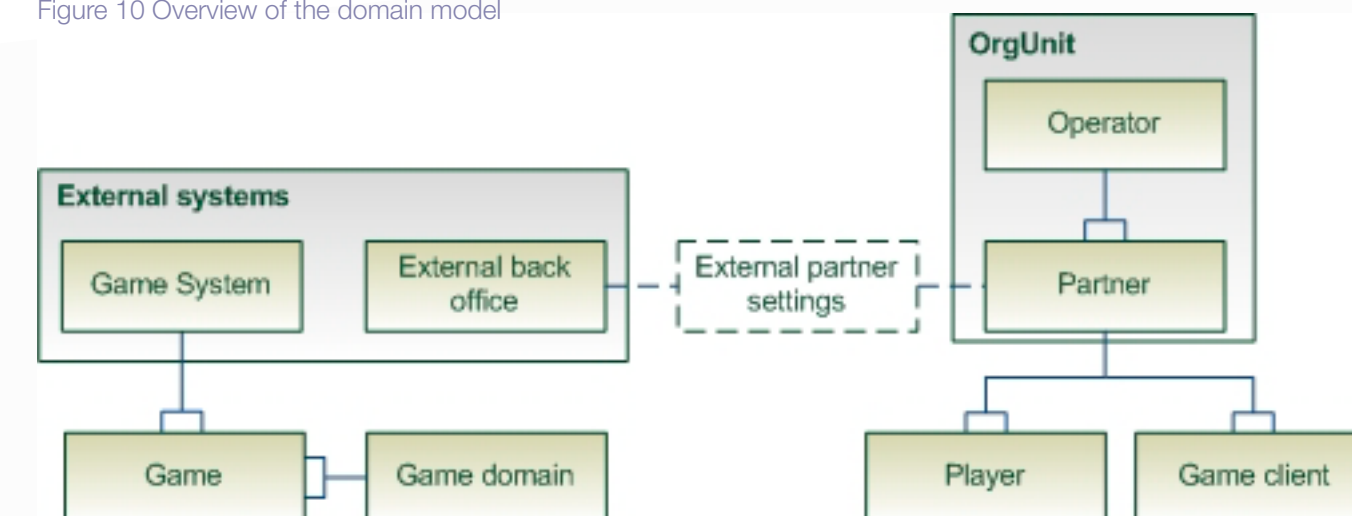
This section describes the technical aspects of the terms operator and partner. For an overview of the GMS organisation, see 2.3 Organisation Overview.

The organisation overview reflects the business model of a GMS operator. However, GMS is built on technology that is dynamic and can be configured to meet the needs of any business. The domain model illustrates the relations between the organisation units from a system point of view and is the foundation of the organisation overview.

The domain model consists of two organisation entities called operator and partner. Please note that these entities are not a direct reflection of any business units but can be used as representations of such. The use of the entities partner and operator is dependent on each specific setup of GMS.

### 3.4.1 Domain Model

Figure 10 Overview of the domain model



### OrgUnit

The OrgUnit is a two tier structure with an operator at the root level or first tier, and the partners as leaves at the second tier. A partner cannot have another partner as a child and all partners are children to the operator at the root level.

The OrgUnit contains common attributes that are inherited by the two entities of the organisation. The common attributes are set when deploying GMS. For example, the operator and partner entity inherit contact information from the OrgUnit.

### Operator

Operator inherits from OrgUnit and represents the root in the organisation, the operator.

### Partner

Partner also inherits from OrgUnit and represents all the partners in GMS.

Partners can be labelled as external (see External partner settings in Figure 10 Overview of the domain model). If a partner is labelled as external, the master account for all players will be located at the external back office, while a shadow account for all players will be established in GMS.

### Player

The player accounts are always connected to a specific partner and the partner can have an unlimited number of players. The players that belong to the partner can only log on to game clients associated with that partner.

### Game Client

A game client acts as an identifier that wraps one or many games from one or many game systems. A game client is always associated with a partner and it is possible for each partner to have any number of related game clients.

A game client is normally used by GMS as an identifier to help the report functionality collect correct data for reports. For example, a partner can have two groups of slot machines, one group with underwater themes and one group with space themes. By grouping them in different game clients the partner has the possibility to get separate reports for each of these groups.

A game client can also be set as a portal; this restricts the functionalities of the game client. A portal does not have permissions to start game sessions or transfer funds. When the game client is set as a portal, it acts as an identifier for portals.

### External Systems

In the external systems illustrated in Figure 10 Overview of the domain model, page 39, only external back office systems and game systems are illustrated. However, all external systems are located in the domain model but only systems relevant for this section are illustrated.

### External Partner Settings

When the external partner settings are set the partner is registered as external in GMS. This means for example that the external player accounts are located in the external back office system.

### External Back Office

To read more about external back office, see section 3.10 External Back Office.

### Game System

A game system is the external system executing the game logic.

From a GMS perspective, a game system is an entity that embraces all games belonging to the game system and which enables the creation of reports based on the included games.

### Game

From a GMS perspective a game is an entity that can be used for game specific reporting. Games can also have administrative attributes such as game vendor.

### Gaming Domain

A gaming domain defines a group of games or game clients, for example poker games or bingo games. A gaming domain can be named to match the included games or game clients. The purpose of grouping games in a gaming domain is to be able to present relevant data from these groups in GMS reports and to be able to separate bonuses from different gaming domains. Games must be added to a gaming domain to be able for use in GMS.

## 3.5 Scalability

The GMS can be deployed in a clustered environment to improve performance, scalability and availability.

A computer cluster is a group of tightly coupled computers that work together so closely that in many respects they can be viewed as though they are a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability.

Most of the processing in multi-user environments like GMS can be done in parallel. A common way to increase performance of tasks that can be parallelised is to use multiple CPUs, resulting in higher perceived performance due to decreased amount of sequential processing. The number of CPUs can be increased in two major ways:

1. Scaling up (vertically) by increasing the capacity of the servers.
2. Scaling out (horizontally) by increasing the number of servers.

### 3.5.1 Scaling up

Scaling up is a straightforward method of achieving scalability. Scaling up is simply to add resources to each node in the system, for example by adding CPU or memory to a single computer.

### 3.5.2 Scaling out

Scaling out is a more cost effective way of achieving scalability, if suitable software architecture with moderate inter-node communication needs is used.

Availability when scaling out is higher in clustered environments as well, because of fail over functionality. If a node fails, its load can be taken over by the remaining nodes, if enough processing capability remains. For this scenario to take place transparently, the software architecture needs to be stateless to avoid losing data and user sessions on the failed node.

For improved performance GMS caches database objects until invalidated by updates or timeout. The cache is clustered so that all server nodes are notified of data changes. Financial data is not cached, to eliminate the risk of using out-of-date information.

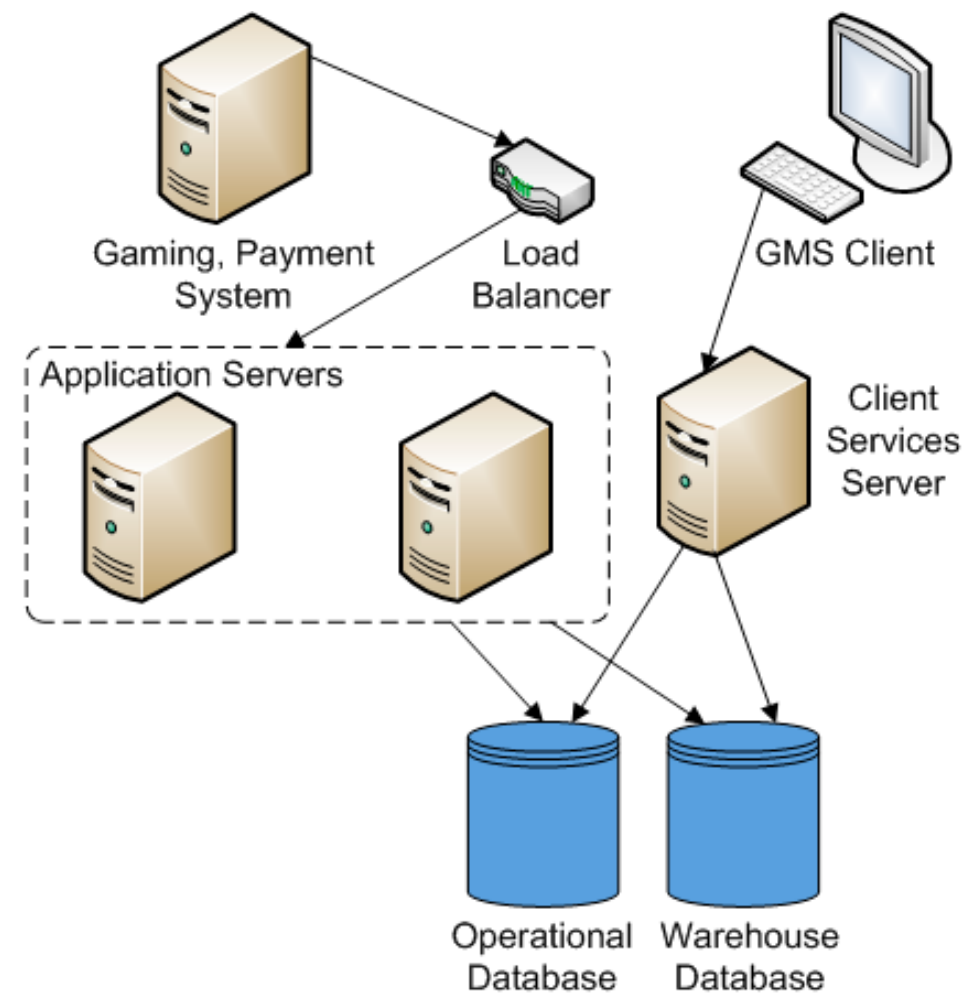
Figure 11 Overview of clustering in GMS

The clusters in an operational environment will typically use a method of clustering with a front-end load balancer and a shared database.

Two types of nodes are available:

1 Gaming and payment services node. There are usually several such nodes in a cluster, as gaming and payment form the major part of the load.

2 Client services node, for serving the GMS client. Usually only a single node is used as the load is not significant.



### 3.6 Security

Boss Media supports different levels of secure connections to ensure that all communication between GMS and the external systems is safely transmitted and not altered or redirected in any way.

#### 3.6.1 Encryption

There are a number of different methods that handle security issues in GMS:

- To hash passwords, the SHA-256 function is used.
- Single Sign-on is used, see section 3.2.2 Single Account / Single Sign-on.
- Session IDs are randomly generated.
- The communication between the GMS client and server is SSL encrypted.
- Individual and group permissions for GMS users.
- Data access permissions.
- RSA integration is available, see section 4.6.2 RSA Login.

#### 3.6.2 RSA Login

The RSA login feature gives a GMS user the possibility to log on with the GMS client from any IP address using an RSA Secure ID device. If RSA Login is to be used, two client services servers have to be setup. One server for IP limited access and the other server for public use with RSA login only.

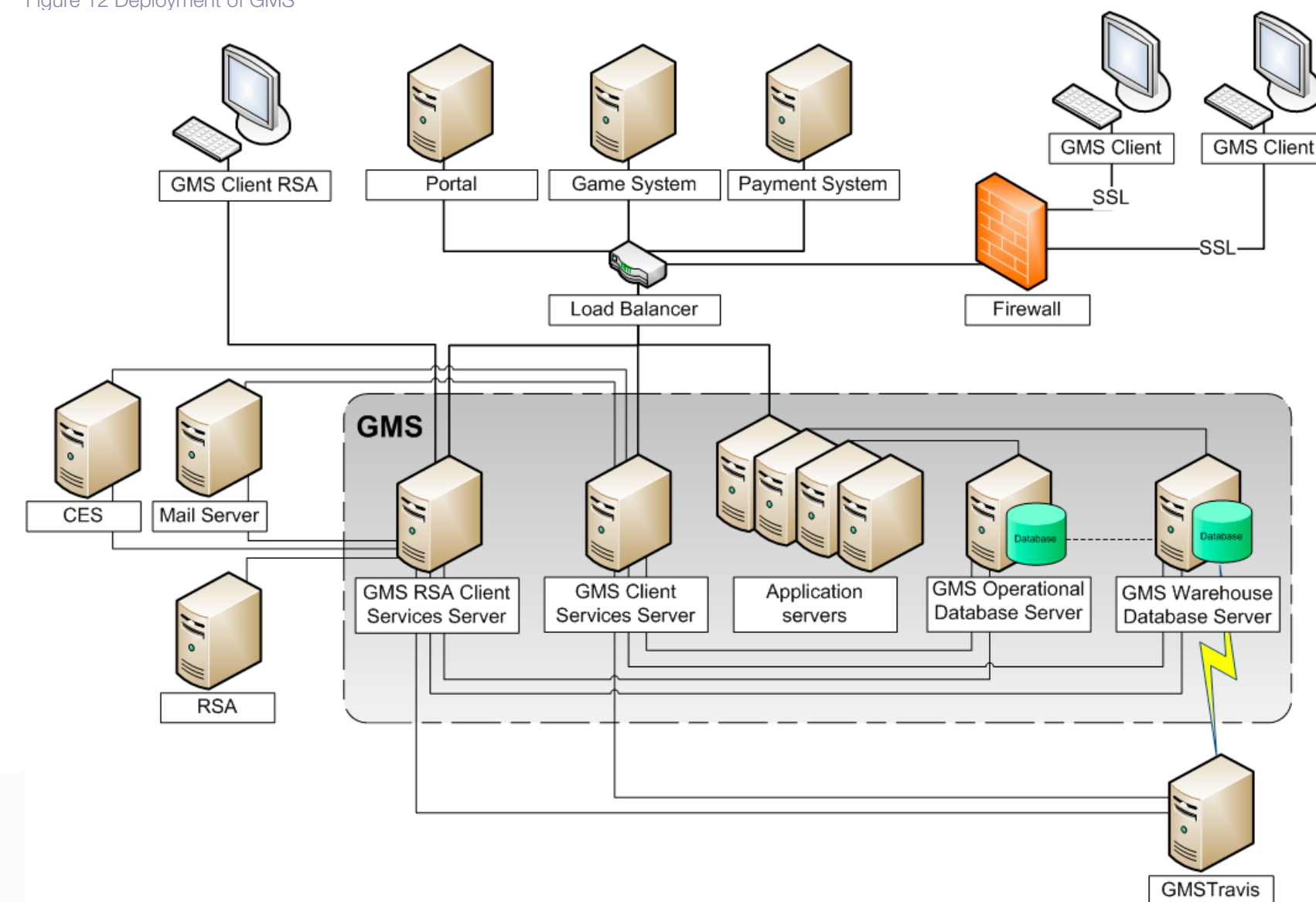
# 4 Deployment

The deployment of the Gaming Management System (GMS) is individual for every setup. The setup depends on user load, data archiving requirements, the games that are deployed, system uptime requirements, and so on. This section describes an example of how to deploy an instance of GMS.

The deployment may be scaled for additional performance requirements regarding application servers and the operational database. The deployment can also be made redundant by adding an extra instance of the components.

In Figure 12 Deployment of GMS you can see how GMS is deployed physically:

Figure 12 Deployment of GMS



For the required software of the GMS client, see 6.4 Hardware and Software Requirements for the GMS Client. Communication is performed using web services over HTTP/HTTPS.

## 4.2 Load Balancer

Calls to the GMS client can be routed through the load balancer on another port number or using direct communication to the server. If SSL encrypted communication is configured, the client, payment/gaming adapter and the SSL traffic is terminated by the load balancer.

The load balancer spreads incoming connections among the members of a cluster. Typically it uses a round robin scheme for distributing the load among gaming nodes.

The load balancer accepts HTTP/HTTPS connections from clients and communicates with GMS with the HTTP protocol.

## 4.3 Application Server

The application server hosts the packaged application and provides resources for the application such as web server, EJB container, database connection pool, web services interface and third party libraries.

The GMS product delivers an optimised version of JBoss.

All application servers and database servers must be time synchronised with each other and handle requests from the game clients. Each application server has one active JBoss/GMS instance.

All JBoss/GMS installations on the application servers must have the same configuration.

GMS on the system services servers will communicate with both the operational database and the warehouse database. A distributed cache is configured in JBoss and a small amount of data is shared between each configured server.

## 4.4 Database

One database server is used as the primary server. Use a separate hot standby server to handle the load if the main server becomes unavailable. Oracle Enterprise Edition must be used on both servers. All data storage is deployed on a SAN. Use separate SAN's for maximum reliability for the main server and the hot standby server.

All database servers and application servers must be time synchronised with each other and handle requests from the game clients.

The recommended database operating system is RedHat Linux or Sun Solaris.

#### **Operational Database**

The operational database is used for “live” gaming, for the player data and for the GMS client operation. GMS uses this schema as its main database, for player data and for player management.

#### **Warehouse Database**

The warehouse database is used for reporting and for data mining. Data is transferred to the warehouse database from the operational database at regular intervals.

#### **Hot Standby Database**

The database system is Oracle Enterprise Edition. A Hot Standby database may be used and if it is deployed use a separate storage. It is recommended to use Oracle Data Guard for the standby server, configured for maximum availability.

#### **4.5 External Back Office**

Deployment of external back office systems are not part of the GMS deployment description, but it should be noted that the level of redundancy and load handling should be equivalent.

## 5 Deployment

### **5.1 Product Technology**

This section describes the technology used in the design of the Gaming Management System (GMS).

#### **5.1.1 Client**

The GMS client is written in C# and runs on the Microsoft .NET Framework platform.

#### **5.1.2 Server**

GMS is a Java Enterprise application developed using standard Java and J2EE.

#### **JBoss**

A preconfigured JBoss is delivered as part of the product. JBoss is responsible for maintaining connections to the databases and managing other resources such as HTTP server, web services, log files, GMS configuration files, JMS queues and distributed caches.

#### **5.1.3 Database**

The database system is Oracle Enterprise Edition.

### **5.2 Hardware Specifications for the GMS Server and Database**

This section contains the recommended hardware that is verified and supported to be used in the deployment of GMS. Other hardware setups can be used and it is then up to the GMS operator to verify the hardware that differs.

#### **Load Balancer/SSL Terminator**

Cisco CSS11503 with SSL modules. These should be in a clustered active/passive setup with transparent failover.

#### **Switches**

HP Cisco Blade GBe enclosure switches. In each Blade Enclosure there should be two switches to ensure redundant performance.

#### **Application Servers**

Application servers should be two HP BL25p G3 or equivalent. The BL25p servers should have 10GB RAM, 2 internal 72GB disks, 2 dual core AMD Opteron CPU's. Required space for an installation is about 350-400 MB for Java (JDK), JBoss and Oracle client and typical memory use is 2GB RAM for each JBoss instance.

#### **Database Servers**

The recommended database servers are HP BL25p G3 with HBA adapters. These servers should be configured the same way as the application servers, however they should have more RAM (typically between 8-16 GB memory) and have an onboard HBA adapter to enable SAN connectivity.

## Storage

Each server should use redundant disk setup. Database servers should have their storage on an HDS 9585V SAN. Dual path configuration should be used where possible to provide high redundancy and performance. The recommendation is to avoid RAID 5 and use RAID 10, because a write operation costs too much performance in RAID 5. Storage requirements are typically between 10 GB and 1 terabyte data/year.

## Report storage server

A dedicated server should be used for storing reports. That is, the reports should not be stored on the GMS server itself. The recommendation is a newer model of HP Proliant server (BL or DL series) with Dual or Quad-Core CPU and 4GB RAM.

## 5.3 Software Requirements for the GMS Server and Database

Recommended software when deploying the GMS database and server:

- Linux (RHEL) or Solaris
- Oracle Database 10g Release 2, Enterprise Edition
- RMAN – Oracle Recovery Manager (<http://www.oracle.com>)
- Java JDK 6 update 4 or later (<http://java.sun.com>)

## 5.4 Hardware and Software Requirements for the GMS Client

Required hardware:

- 512 MB RAM
- 100 MB of disk space
- 1 GHz CPU
- A minimum resolution of 1024 x 768

Required software:

- Operating System:
  - Microsoft Windows 2000 – Not verified
  - Microsoft Windows XP – Supported
  - Microsoft Windows Vista 32-bit – Supported
  - Microsoft Windows Vista 64 bit – Supported
  - Microsoft Windows 7 – Supported
  - Microsoft .NET Framework 2.0 or later

Recommended software:

- Microsoft Excel 2003 or later
- Adobe Acrobat Reader (<http://www.adobe.com>)

# 6 Terms and Acronyms

## 6.1 Terms

The following terms are used in the document.

Terms	Description
<b>Adapter</b>	An adapter is a self contained component which can be designed to use the most suitable protocol to communicate with an external system.
<b>Base Currency</b>	GMS has a single defined base currency. All financial transactions are stored in both the currency in which the transaction was carried out and the base currency.
<b>Bonus Bag</b>	The container of bonus money.
<b>Bonus Money</b>	Bonus money is money that carries some sort of requirements that must be fulfilled before it can be converted into real money, for example wagering requirements. Bonus money cannot be withdrawn.
<b>Cashier</b>	The interface from where the player can manage deposits and withdrawals to and from the gaming system.
<b>Cluster</b>	A group of tightly coupled computers that work together so closely that in many respects they can be viewed as a single computer.
<b>Deposit</b>	A player makes a deposit when transferring real money into his/her real money account from a third party payment system or a third party back office system. A deposit is credited to the player's real money balance.
<b>Direct Purchase</b>	When the game requests money from the central player account, only enough to cover the actual bet.
<b>External System</b>	An external system is a system that GMS can integrate with.
<b>Frequent Player Level</b>	The Frequent Player Level can be used as a tool for sorting players into categories of your choice. A game a player can participate in. A game has two parts, a server part that runs on the game server and a client part that is wrapped into a game client.
<b>Game</b>	A game client is an entity that wraps one or many games from one or many gaming systems. The external system executing the game logic.
<b>Game Client</b>	A logical grouping of games. For example Casino, Poker, or Bingo.

<b>Terms</b>	<b>Description</b>
<b>Game System</b>	The external system executing the game logic.
<b>Gaming domain</b>	A logical grouping of games. For example Casino, Poker, or Bingo.
<b>Gaming Platform (GP)</b>	Boss Media's general game system for different games such as instant online casino games and instant online bingo games.
<b>GMS</b>	The Gaming Management System (GMS) is a Boss Media product that provides comprehensive integration and administrative back office functionalities.
<b>GMS Instance</b>	A single set-up of the Gaming Management System (GMS).
<b>GMSTravis</b>	A Boss Media tracking system that GMS can integrate with.
<b>Gross Gaming</b>	Gross gaming is the difference between all money bet and all money won (including bonus money and jackpot).
<b>Hibernate</b>	An object-relational mapping library for the JAVA language.
<b>Jackpot</b>	A grand prize in gaming. A small sum of every bet is placed in the jackpot, continuously increasing the sum of the jackpot. When a jackpot pays out, the jackpot is set to zero and the process starts again.
<b>Java</b>	A programming language developed by Sun Microsystems.
<b>JBoss</b>	JBoss Application Server (or JBoss AS) is a free software/open source Java EE-based application server. Because it is Java-based, the JBoss application server is usable on any operating system that Java supports.
<b>Load Balancer</b>	A load balancer distributes the load between resources in a system.
<b>Lobby</b>	A virtual location where players can select and launch games.
<b>Loyalty Points</b>	Each player has a loyalty point account. Loyalty points in the account are accumulated when the player wagers real money in games. Loyalty points can be converted to real money.
<b>Node</b>	A device connected to a network, such as a computer or a router.

<b>Terms</b>	<b>Description</b>
<b>Pending Bonus Bag</b>	A bonus payment that the player receives when meeting the qualifying requirements.
<b>Pending Loyalty Points</b>	Loyalty points that the player receives when the player meets qualifying requirements.
<b>Portal</b>	An interface that the player can use to launch wanted games, that is, a web portal that is a collection of services offered to the players.
<b>Qualifying Requirement</b>	An amount that states how much a player must wager before qualifying to receive a bonus payment. Based either on bonus amount or bonus + deposited amount.
<b>Real Money</b>	Real money that the player can wager with.
<b>Redundant</b>	When a system's critical components is duplicated to increase the reliability of the system.
<b>Reward</b>	An amount of bonus money or loyalty points that the player receives when meeting certain requirements of a promotion.
<b>Reward Abuser</b>	A player which systematically takes advantage of the available rewards of a game system in a not allowed manner.
<b>RSA</b>	RSA is an algorithm for public-key cryptography.
<b>Servlet</b>	A Java programming language class.
<b>Shadow Account</b>	A mirrored version of the player account.
<b>Shadow Balance</b>	The last known account balance of an ongoing player session.
<b>Single Sign-On (SSO)</b>	Single sign-on (SSO) is a method of access control that enables a user to authenticate once and gain access to the resources of multiple software systems.
<b>Spring</b>	The Spring Framework (or Spring for short) is an open source application framework for the Java platform.

<b>Terms</b>	<b>Description</b>
<b>Total Win Amount</b>	The amount of real money and bonus money the game returns to the player including jackpot wins.
<b>Transfer to Lobby</b>	A real money or bonus money transfer from GMS to a game lobby.
<b>Wagering Requirement</b>	An amount that states how much a player must wager before the bonus money is converted to real money.
<b>Web Service</b>	A web based service designed to support interaction between entities in a network.
<b>WebDollar</b>	A Boss Media payment system that GMS can integrate with.
<b>Withdrawal</b>	A transfer of real money to a payment system from the player's real money account located in GMS.

The following acronyms are used in the document.

<b>Acronyms</b>	<b>Description</b>
<b>API</b>	Application Programming Interface
<b>CES</b>	Currency Exchange Server
<b>EBO</b>	External Back Office
<b>EJB 3</b>	Enterprise Java Bean version 3
<b>GDS</b>	Gaming Data Service
<b>GMS</b>	Gaming Management System
<b>GUI</b>	Graphical User Interface
<b>JMX</b>	Java Management Extensions
<b>O/R</b>	Object/Relational
<b>RMI</b>	Remote Method Invocation
<b>SOAP</b>	Simple Object Access Protocol
<b>WSDL</b>	Web Service Description Language
<b>WYSIWYG</b>	What You See Is What You Get
<b>XML</b>	Extensible Markup Language