Mobile Games Interoperability Forum

Mobile and Wireless Gaming White Paper

Version 1.0

Abstract:

The wireless gaming market is heterogeneous and fragmented because of existing technology barriers and missing interoperability. The Mobile Game Interoperability Forum (MGIF) is an open industry forum founded by Ericsson, Motorola, Nokia and Siemens to establish an industry wide standard to enable the market. The MGIF will define mobile games services, interoperability specifications and application programming interfaces (APIs) for network-based servers for the benefit of all stakeholders.

This white paper describes the MGIF goals and approach and the associated relations to technologies and other activities.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

Table of Contents

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

1. THE FUTURE OF WIRELESS GAMING

The convergence of telecommunications, media and information technology enables new kind of services that can be used anywhere, anytime and with any device. The increase of network capacity enables rich media on mobile terminals. Mobile terminals will become more and more advanced. Different modes of wireless communications will be integrated, from indoor networks, such as WLAN and Bluetooth, to cellular signals, radio and TV broadcasting, and satellite communications. Mobile devices will roam freely from one standard to another. Entertainment will be a key revenue generator for all stakeholders in the mobile business. The MGIF specifications address the issues of portability and interoperability in the mobile games space. The MGIF specifications will allow game developers to produce and deploy mobile games that can be more easily ported between multiple MGIF platforms and wireless networks. Games can be played over different mobile devices. The intent is to produce significant cost reduction for game developers and game platform owners. While the MGIF will not specify functionality for mobile devices, it will interact with appropriate standards bodies and forums to enable an open end-to-end games environment. The MGIF is defining API specifications; it will not produce a mobile game platform.

The potential scope of a gaming platform is enormous. Therefore a pragmatic approach has been taken where initial efforts have been concentrated in those areas considered to reap the most benefit for the game developer. Specifically, the following areas are addressed in the initial release:

Metering, which relates to how the game is charged for and consequently how revenue is collected;

Connectivity, with the intent to reduce the complexity of interfacing through different messaging systems;

Session Management, which provides the core event-based framework upon which a game can be built;

Score and Competition Management, which provides the basis upon which online communities can be built in the mobile gaming domain; and finally

Logging, which attempts to standardize system reporting and consequently reduce game management and hosting costs.

The MGIF Platform Specification v1.1 release, which is available on 21 October 2002, represents an industry milestone. Key industry players have reached a consensus on both the issues to address in the standardization efforts and the technical approach to be used. Over the next few months the MGIF will be building on this foundation to increase both the depth and the breadth of this specification.

2. MOBILE AND WIRELESS GAMING TECHNOLOGY TRENDS

The expression “mobile gaming” means a user is able to play games anywhere and anytime with the access device he/she is carrying. “Wireless communication” indicates that no physical connection exists between the sender and the receiver. The difference between the terms “mobile” and “wireless” is as follows: Wireless technology is not always fully mobile but the coverage may be restricted to a certain geographical area. In this paper the terms *wireless* and *mobile* are used as synonyms. The expression *mobile gaming* is used for the domain and the expression *mobile termina*l refers to the access or client device.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

Mobile games can be built into the client device, downloaded to the device, or accessed via networks. The focus of the MGIF is on games that access a game server during play. The MGIF Terminal Requirements committee addresses the needs of game developers regarding mobile device features and functionality. For these games the development of network infrastructure and client side technologies provide new opportunities and challenges.

Technical improvements drive seamless client-server communication in any application. These include increases in bearer network bandwidth and capacity increases. Instant bearer network connection and new solutions for low range wireless connections make networking ubiquitous for the client. Chapter 3.1 describes the development of the network infrastructure in more detail. There is an increase in the variations between different handsets with regard to display size, input mechanisms and functionality. The selection of different add-on functionality, including radio, digital camera, MP3 player or thermometer, transforms a mobile phone into a multi-purpose device. Traditional phone-only handsets are becoming PDA-like devices containing personal information management utilities such as calendars, memos and address books as well as entertainment options. Mobile phones have also become a compulsory fashion item needing frequent updates.

On the client side, there are three main streams evolving to enhance the user experience in mobile gaming: messaging, browser based technology exploiting Internet investments, and devices having a runtime environment for executable games. Please see Chapters 4.1 - 4.3 for more information.

3. INFRASTRUCTURE

3.1. NETWORK INFRASTRUCTURE

GSM (Global System for Mobile Communication) is the most widespread 2G wireless standard. Packet data is introduced into GSM systems by using GPRS (General Packet Radio Services). The main effect of GPRS for game developers is that the dial-up process is removed. The start of an application or resuming it after an interrupt is instant. GPRS is implemented on top of the existing infrastructure and is an intermediate step towards 3G systems, which makes it an economically sensible investment for GSM operators. The next step toward full-scale 3G services is either EDGE (Enhanced Data rates for GSM Evolution) or WCDMA (Wideband Code-Division Multiple Access) with or without EDGE. EDGE concentrates on improving the capacity and efficiency of the air interface. WCDMA or UMTS (Universal Mobile Telecommunications System) introduces a completely new radio interface, meaning that the network infrastructure has to be rebuilt from scratch.

TDMA (Time Division Multiple Access), cdmaOne and PDC (Personal Digital Cellular) have their own migration paths leading to 3G-like networks with similar enhancements as in the GSM systems. For TDMA, EDGE is a natural step. Cdma2000 is intended to be more advanced than WCDMA in its phase one. In Japan the migration from PDC to WCDMA has been somewhat delayed from the original estimates but will be faster than in Europe.

In spite of these improvements game developers cannot expect their users to gain much higher bit rates under all conditions. For example the number of users affects the performance of packet data networks significantly. Rural areas will be limited to EDGE or GPRS coverage for many years. The nature of CDMA systems makes the bit rate highly dependent on the distance from the mobile terminal to the base station. The QoS (Quality of Service) classes are intended to guarantee optimal conditions for network communications for different kinds of applications. Pricing policy may cause game players to choose a moderate service level that must be taken into account in the design of the game.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

Bluetooth and WLAN (Wireless Local Area Network) both enable short-range connectivity that creates an opportunity for new kinds of applications based on location awareness and restricted access, where security aspects may apply.

4. TERMINAL APPLICATION PLATFORMS

Clients vary widely and include message based, browser based and executable types.

4.1. MESSAGE CLIENTS

Mobile messaging is evolving beyond text by taking a development path from SMS (Short Message Service) to EMS (Enhanced Messaging Service) and MMS (Multimedia Messaging Service).

EMS facilitates screensavers, logos, pictures and ring-tones. Not all device manufacturers support all of the EMS capabilities, for instance Nokia offers a proprietary solution, Smart Messaging™, for similar purposes.

MMS provides the ability to send messages comprising a combination of text, sounds, images and video to MMS capable handsets. All the major handset manufacturers have adapted MMS. A new mobile network infrastructure is needed for MMS.

4.2. BROWSER CLIENTS

There are two basic approaches for browser clients: WAP (Wireless Application Protocol) and HTTP clients. WAP is a suite of protocols optimized for wireless communications. For the application developer, clients can be differentiated by the markup language used, as the WAP gateway converts WAP protocols to wired Internet protocols and vice versa. Current WAP micro-browsers render WML but the trend is to support XHTML (Extensible Hypertext Markup Language), which will be the universal language for wireless and wired browsers.

Markup languages that are used in wireless context are:

- HTML
- WML (there are two inconsistent browser implementations: UP.Browser by Openwave and Nokia browser)
- XHMTL (which appears to be the future trend)
- HDML –Handheld Devices Markup Language (Openwave, slowly dying out)
- c-HTML – Compact HTML (NTT DoCoMo/i-mode)
- MML – Mobile Markup Language (J-Phone/J-Sky)

The main enhancements of WAP 2.0 from the game developer's point of view are:

- WML2 based on XHTML, WAP Application Environment supporting CSS
- Enhanced push functionality which allows applications to send notifications to their users
- Enhanced User Agent Profile (UAProf) service based on CC/PP, providing a mechanism for describing the capabilities of clients and the preferences of users to an application server
- Persistent Storage Interface
- Multimedia Messaging Service (MMS) which facilitates the mixing of messaging and browsing

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

- Support for Pictograms i.e. tiny images, which can be used as symbols to save screen space.

4.3. EXECUTABLE CLIENTS

Executable games facilitate applications with components that execute directly on the device. Local processing and storage make a real-time gaming experience possible because network latency can be eliminated. Compared to desktop and notebook computers, the mobile terminal offers a limited executing environment due to battery capacity, processing power, small display size and a limited keyboard. These limitations have to be taken into account for specific design considerations.

Most of the companies in the business have chosen Java™ technology on the server side. Currently Java™ is also the main choice on the client side.

MGIF has chosen the Java 2 Platform, Micro Edition (J2ME ™) to be the focus technology for executable clients. J2ME devices allow the download and run of Java applications called 'Midlets'. These applications can be downloaded either from a PC or from the Internet via wireless communication.

Solutions for executable clients in mobile terminals are:

- Java-based
  - J2ME that fulfills the CDLC/MIDP specification
  - I-appli (NTT-Docomo)
  - Personal Java
  - ExEn
- C/C++ and Java
  - EPOC (Symbian)
  - BREW (Qualcomm)
  - Win32 (Windows CE)
- CLR (Common Language Runtime) and the .NET™ framework (Microsoft)

Game consoles as potential clients and the technologies involved with them are not discussed herein. Those technologies will have significant impact on this area in future.

5. MOBILE GAMES INTEROPERABILITY FORUM (MGIF)

The MGIF is an industry forum founded by Ericsson, Motorola, Nokia and Siemens. The MGIF will define mobile games interoperability specifications and application programming interfaces (APIs) for network-based servers. The specifications will enable game developers to produce, deploy and manage mobile games that can be distributed across multiple game servers and wireless networks, and played over different mobile devices.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

5.1. KEY VALUES OF MGIF

Provide consistent platform for mobile games

By defining a baseline set of assured available services, developers are provided with consistent functionality, which in turn adds certainty to the market and saves developers the expense of replicating potentially redundant functionality.

Reduce cost of development

Game developers can change service provider with minimum of porting costs because of a standardized access to gaming platform services through public APIs.

Reduce cost of ownership

Game hosts and/or carriers can administer multiple games from multiple sources with minimum setup and maintenance costs because games and game platform are presented interoperable standard interfaces.

Enable new technologies/new business

By addressing core network interoperability issues, MGIF intends to enable new technologies to be exploited and innovative business models to be successfully implemented that would have taken too long to naturally evolve in the marketplace. (e.g. enabling cross carrier gaming by providing APIs that mediate connectivity and revenue sharing)

6. MGIF BENEFITS

The MGIF concentrates on removing technological barriers and thus increasing interoperability in wireless gaming. Once adopted, the MGIF specifications will benefit a number of stakeholders. The most important benefit for all stakeholders is the extra boost open standards give to the mobile gaming industry. Individual industry stakeholder benefits are summarized below.

6.1. GAMES DEVELOPERS BENEFITS

By developing MGIF compliant games, developers get immediate access to a wider customer base once their games are launched. By facilitating games and platform interoperability the MGIF improves developers' productivity by enabling a game to be ported more easily to any MGIF compliant gaming platform (MGIF Portable – see below). This avoids locking developers to platform vendors. MGIF specifications and design guidelines reduce development work, cost and complexity. Using MGIF specifications, developer can focus on the game logic and user experience, relying on the defined interfaces to supply the underlying implementation details.

6.2. PLATFORM DEVELOPERS BENEFITS

The MGIF specification provides a well-defined set of APIs that game platform developers have to implement in order to be MGIF compliant. By leveraging the MGIF specifications to improve platforms interoperability and games portability, platform developers will be able to attract a wider game developer community to develop attractive content for their platform.

6.3. SERVICE PROVIDERS BENEFITS

Service providers consider game services a future key revenue generating service.

Because games will run on any compatible MGIF compliant gaming platform, MGIF specifications enable wireless service providers to choose the best game content. MGIF

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

specifications will mitigate the reliance between the gaming platform and the content, hence avoiding vendor lock.

6.4. NETWORK OPERATORS BENEFITS

Network Operators provide mobile games as value-added services to their customers. As end users use various access devices and methods to interact with these games, Network Operators expect to be able to leverage these mobile games in order to help drive growth in the space of Seamless Access.

MGIF specifications ensure that various platform vendor solutions will interoperate, reduce the risk of vendor lock and increase the selection of available games. MGIF accelerates the development of new mobile games that will showcase the next generation networks and handsets.

6.5. HANDSET MANUFACTURERS BENEFITS

Handset manufactures are seeking applications and content to motivate consumers to use new commercialized inventions, thus affecting their wide acceptance. By simplifying the development of mobile games and leveraging the synergies between server and hand-sets, the MGIF specifications facilitate wide exploitation of new devices, infrastructure and software products by allowing the development of new products and adding support for new technology into the existing solutions.

6.6. IDE PROVIDER BENEFITS

Through the adoption of MGIF specifications, a single development tool can support a range of platforms, increasing the potential market for the tool.

6.7. CONSUMERS BENEFITS

Upon wide acceptance, MGIF specifications increase the offering of games that are interactive, encourage participation, and have community aspects. MGIF specifications also facilitate consumers' access to a variety of games irrespective of their individual choice of device and operator.

7. MGIF DEFINITIONS OF PORTABILITY AND INTEROPERABILITY

Standards for portability and interoperability are needed to boost the mobile gaming business. Alack of common standards creates additional development work and adds complexity to games and gaming platform developers.

Portability refers to the software's ability to run on more than one platform without major modification.

In MGIF terms, portability is the ability of a game to run on any MGIF compliant platform while requiring minimal changes to its code. The MGIF APIs enable a degree of games portability using the same development language, game execution environment and a minimal use of platform specific APIs.

In Figure 1 we show the basic principles of games portability. Given the assumptions named above, if game "x" is developed according to the MGIF specifications, it is able to execute in both Environments with minimal changes to its code.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

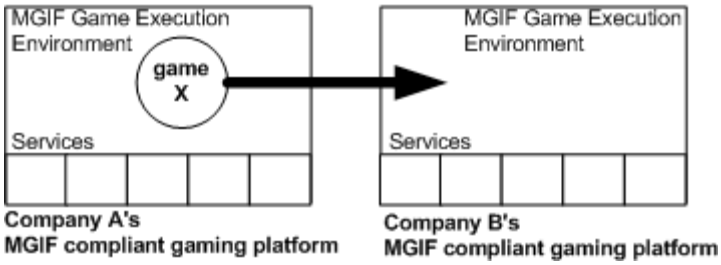

Figure 1. Portability

Interoperability refers to the ability of applications running on distributed systems architecture to exchange information and share services in a cooperative fashion.

In MGIF terms, interoperability is the ability of MGIF compliant games and platforms to communicate and share resources and services. The abstraction of the environment enables mapping of multiple logical resource instances onto one or more physical resources in a transparent manner. Further, this abstraction allows composing basic gaming services to form more sophisticated games. The MGIF addresses interoperability requirements in terms of both the definition of services interfaces (APIs) and the identification of protocols to invoke a particular interface.

Figure 2 shows how two games running on different game execution environments are able to exchange information and access each other's gaming platforms services in a seamless manner.

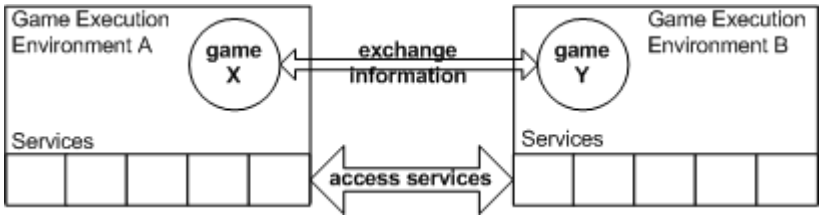

Figure 2 . Interoperabilty

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

8. MGIF CONCEPTUAL ARCHITECTURE

The MGIF Architecture for a game platform consists of three separate layers, as illustrated in Figure 3. These are the *connector*, *game execution environment* and *services* layers. Please notice that any detailed service categories and their names in figure 3 are provided only to clarify the architecture and should not be considered mandatory.

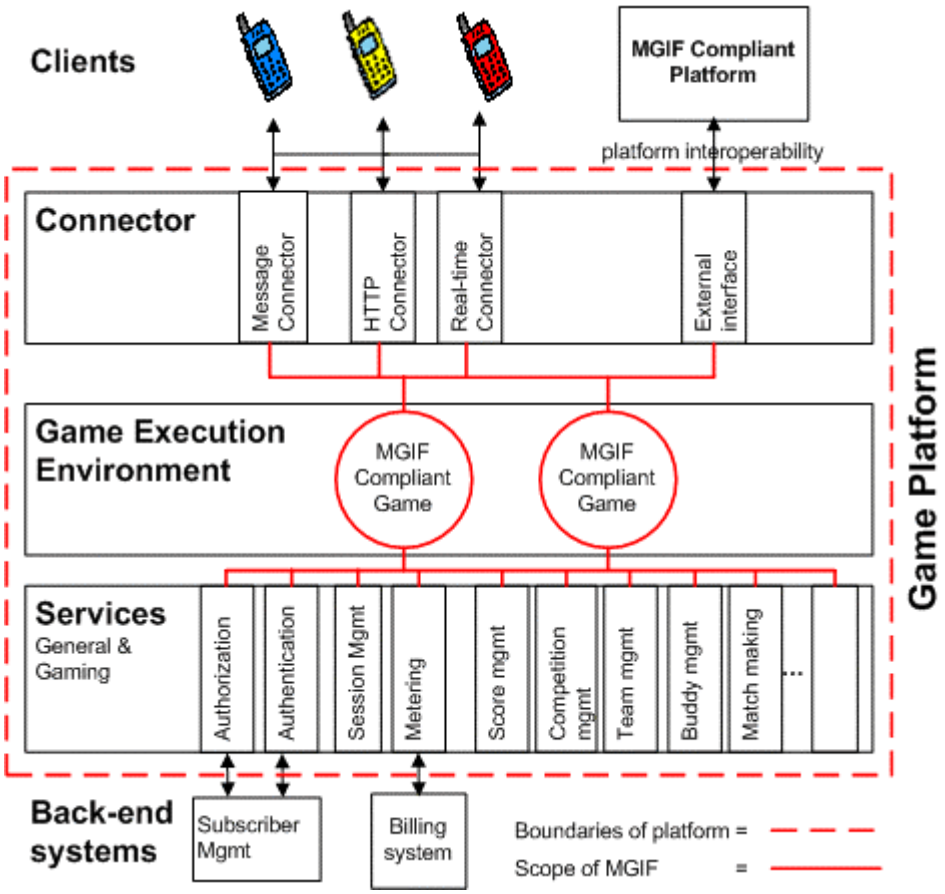

Figure 3. MGIF Architecture.

8.1. CONNECTORS

The **connector** is a mediator between the mobile terminal and the gaming platform. It hides the intricacies of different client network protocol implementations and terminal characteristics. Different connectors are defined the suit the needs of different client types, for example, WAP games require an HTTP connector whereas SMS games require a message connector hiding SMSC specific protocols.

The connector transforms a protocol dependent message into an MGIF defined message and passes it to the Game via the Game Execution Environment. The connector also parses available information about the mobile terminal and the user from the incoming message.

The connector transforms outgoing MGIF messages into the format and protocol required by the particular carrier network and mobile terminal.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

8.2. GAME EXECUTION ENVIRONMENT

Games run within the **Game Execution Environment**. The Game Execution Environment is always specific to an underlying runtime environment offering a number of services and APIs that are directly or indirectly used by the game applications.

The Game Execution Environment offers game developers interfaces to generic and game specific services. MGIF only defines interfaces for accessing these services leaving all implementation issues for the platform developers. This way, MGIF defined interfaces hide all implementation intricacies proving homogeneous access to heterogeneous services that can be differently provided by operators and service providers.

8.3. SERVICES

Services are either generic or game specific.

Generic services include application and user management, billing, load balancing and metering.

Game specific services provide features for game applications. MGIF standardizes the interfaces for these game specific services.

As examples, gaming services could include:

- **Score Management** - store scores and results
- **Competition Management** - organize competitions and tournaments
- **Team Management** - handle games that are team based (players are part of a team)
- **Buddy Management** - create teams based on buddies, chat with buddies, match making with buddies
- **Match Making** - selects one or more players (or teams) to play a game on given conditions.
- **Presentation** – including messaging and markup.

9. MGIF GAME CONCEPTS

MGIF recognizes the following high level game related concepts, amongst others.

9.1.1 Award

The MGIF recognizes these concepts for **Award**:

- Award is a reward a player or a team is entitled to, either as a concrete prize, as a position in the Hall of Fame or as a qualification for the next round;
- A player or a team entitlement to an award is defined based on score or ranking;
- Once awarded, players or teams may be excluded from a game, for example for a certain time frame.

The notion of Award is optional for an MGIF compliant game.

9.1.2 Community

The MGIF has the following concepts for **Community**:

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

- A Community is a group of players registered in the system to play together, typically against each other;
- A Community has to be registered, otherwise it is not a community from the system's point of view;
- Regarding identification communities are similar to players.

An MGIF compliant game may but is not required to support the notion of Community.

9.1.3 Contest

The MGIF concepts for **Contest** are:

- A Contest is a structural description of an entertainment concept involving one or more games;
- Contests have a time frame in which they are active i.e. playable. When game lists are accessed via contests, only active games are shown;
- Contests may have rules that control who can participate or win. When game lists are accessed by a player via contests, only games accessible to that particular player are shown;
- Contests may consist of phases that are contests;
- Contests involve an award that may be a qualification for the next phase;

An MGIF compliant game may but is not required to support the notion of Contest.

9.1.4 Game

The MGIF concepts for **Game** are:

- Game is a concept of entertainment implemented as a game application and installed for a particular content provider/service provider (customer);
- Games must have a reference to be accessed. The reference is client type dependant e.g. a keyword for SMS or URL for browsing;
- Games have a description that can be used in game listings;
- Games can have a price, subscription, registration or other conditions to authorize play;
- Games are managed through a life cycle that defines how they are loaded, instantiated, joined by players, played by registering gaming events and ended;
- It is recommended that the game provide an interface for online monitoring of game instances;
- Games provide information for statistics that is based on game instances;
- Games can be used to create contests, if contests are supported.

An MGIF compliant game must support the notion of Game.

9.1.5 Game Instance

The MGIF concepts for the **Game Instance** are:

- Game Instance is the game that is running and is currently played by zero or more players.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

An MGIF compliant game must support the notion of Game Instance.

9.1.6 Gaming Event

The MGIF concepts for **Gaming Event** are:

- A Gaming Event is a measurable step in the game that is recorded;
- Gaming Events are the basis for metering, used for game statistics and/or billing;
- The result of a Gaming Event may affect to the score of a player;
- A Gaming Event type has a unique identification which are configured to match the gaming event classification specified by MGIF;
- Gaming events can be categorized into tariff classes;
- The recorded gaming events can be retrieved using criteria consisting of: time period, event type identification, game, game instance, player, team (if supported) or contest (if supported).

An MGIF compliant game must support the notion of Gaming Event.

9.1.7 Player

The MGIF concepts for **Player** are:

- A Player is an individual user that has once played or is currently playing a game;
- A Player can instantiate games or join existing game instances
- A Players may be a registered user, but is not required to be so;
- A Player must have a unique identification in the system that can be associated with the player's results;
- Unique player identification has to be persistent in time for as long as the contest lasts;
- Unique player identification does not have to be necessary visible but it can be;
- Players have a visible nickname that is shown in the results list. The nickname does not have to be unique within the system;
- Players may have additional contact information and data related to playing such as a role or an avatar. These are out of scope of MGIF unless they affect scoring.

An MGIF compliant game must support the notion of Player.

9.1.8 Ranking

The MGIF concepts for **Ranking** are:

- Ranking is the relative position of a player or a team when compared to other competitors;
- Explicit ranking is an alternative to score based ranking.

An MGIF compliant game must support the notion of Ranking.

9.1.9 Score

The MGIF concepts for **Score** are:

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

- A Score is a quantitative result of a player or a team;
- Scores are recorded either during the game or after its completion;
- Scores can be scaled using an algorithm such as a scaling factor in order for scores to be comparable across games;
- Scores can be retrieved and listed using criteria similar to those for gaming events (see 9.1.6 above);
- Scoring logic within a game is out of scope for MGIF.

An MGIF compliant game must support the notion of Score.

9.1.10 Team

The MGIF concepts for **Team** are:

- A Team is a group of players united to play as a single unit in competition with other teams;
- A Team must be registered. Otherwise it is not a team from the system's point of view;
- Teams' identifications are similar to players' (see 9.1.7 above);
- Teams consist of n players or up to n-1 teams where n is the total number of players in all included teams.
- Participation of a team may be restricted to a community, if communities are supported.

An MGIF compliant game may, but is not required, to support the notion of Team.

10. MGIF SCOPE

The aim of the MGIF initiative is to specify an industry wide standard and to develop certification procedures to encourage wide adoption of the standard. The MGI Forum will work with and provide recommendations to the appropriate standards bodies and forums to ensure an open environment for developing mobile games.

The MGIF contributions to the mobile gaming business are:

- The concept of a gaming and gaming platform architecture blueprint common to various kind of games;
- Abstraction to hide the heterogeneous world of communications and terminals;
- Specification of the gaming services used by games to offer a specific functionality and services used to provide games to the user community;
- Interface specification of the services and their communication with the bearer and device layers as well with the service providers or third party backend systems;
- A reference implementation and SDK for the programming languages of choice. The first choice is Java ™.

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

11. MGIF OUT-OF-SCOPE

The MGIF does not define specifications regarding the gaming experience itself, which remains the distinctive feature of games, encouraging competition among game developers.

The MGIF depends on other initiatives developing infrastructure and specifications for the Internet and the mobile communications industry. MGIF will not reinvent the wheel, nor try to specify general-purpose solutions even if the industry is lacking a fully satisfactory solution. Examples of issues that are out of scope are a completely device independent GUI framework or a roaming solution between different operators and countries.

12. MGIF V1 SPECIFICATIONS DELIVERABLES

Session Management:

This specification contains abstractions of the MGIF Gaming Platform's object model and its representation in Java™ classes and interface terms and the abstraction of the lifecycle of game sessions and all possible gaming events. This specification allows game developer to design game logic accordingly to MGIF game concepts. Game developers may use hidden MGIF execution environment services, e.g. services that have no direct API calls. These "non-callable" services include such facilities as input routing to a specific game session, execution of hooks as reactions to lifecycle events, use of timers, and maintenance of game session objects' encapsulation and identification.

Connectivity:

The connectivity APIs enable communication between the game and the clients. The connectivity APIs specify how the requests from clients are exposed to the games, and how applications generate responses to the clients.

Metering:

The metering API allows the game application to post metering events at key points in its execution, e.g. when a new level is entered or when some optional resource or feature is selected. The MGIF platform can use this stream of events provided by the application for both informational logging and to implement billing.

Scores & Competition Management:

The scores & competition managementAPI allows the game application to record scores in the available score tables, and to retrieve previous scores in a variety of manners. This API allows competitions of various sorts to be implemented either within the application itself, or behind the scenes in the MGIF platform.

Timers:

The specification contains the functionality for the creation and behavior of timer based game session lifecycle events. A game developer is able to write game logic that creates, cancels and reacts on timer-based events. This ability is sufficient for development of games that are based on more than reaction to players' inputs.

Logging:

The logging APIs enable recording of logging information from a game to game platform.

Logging information can be used from a service provider perspective to provide two important functions:

- To troubleshoot in the case of malfunctioning of badly behaving games;

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

- To gather data for data-warehousing, and data mining purposes.

13. MGIF API SPECIFIC USE CASES AND BENEFITS

13.1. GAME DESIGN

13.1.1 Use Case: Current Status

Game developer A designs a game to be implemented against platform X. Later the requirement changes and the game must be moved to platform Y. This requires the design to be reworked from scratch since the concepts in the two platforms are different.

13.1.2 Use Case: MGIF Benefit

Game platforms X and Y are MGIF compliant and consequently share the same basic game concepts. The majority of the game design can be reused.

13.2. CONNECTORS

13.2.1 Use Case: Current Status

Game Developer A is developing an SMS based game to be hosted by Game Platform Provider A for Mobile Operator A. Game Platform Provider A has a propriety SMS API that Game Developer A has to develop for. The propriety SMS API requires time and effort to learn and implement.

Mobile Operator A has a change of strategy and now uses Game Platform Provider B for its games platform. Game Platform Provider B uses a different proprietary SMS interface. Game Developer A now must undertake the work to rewrite all the connectivity aspects of its games.

Mobile Operator B wishes to license the same game but for WAP rather than SMS. Game Developer A now must completely rewrite of the game code for the new media.

13.2.2 Use Case: MGIF Benefit

Game Platform Provider A and Game Platform Provider B have MGIF compliant platforms that both offer the MGIF connectivity API to their developers. This is a simple API that requires minimal effort to learn.

When circumstances require a change of game platform, the connectivity aspects of the games do not have to be reworked, as the API supported by each platform is identical. The MGIF connectivity API abstracts basic input and output away from the actual medium. The result is that a WAP application can be exposed quickly and easily using the same framework that was used for the SMS game. The developer can still augment the game to make use of the more sophisticated interface that WAP provides, but the base functionality can be transported with minimal effort.

13.3. METERING

13.3.1 Use case: Current Status

Game Developer A and Game Developer B both develop games they wish to publish as widely as possible. The two primary carriers, Mobile Operator A and Mobile Operator B, charge their

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

customers in very different ways for game play. Mobile Operator A charges per month of use while Mobile Operator B charges a fixed amount for 10 games.

In the current scenario, the games developers have to rework their games considerably to enable the differing charging models. This may be further complicated by the Games platform that may or may not provide APIs that support the required business models.

The responsibility for who decides the price that is attached to certain “events” may vary. In some instances the game developer will be involved, in some instances this function will be provided by the game platform company and in others the carrier will take this responsibility directly. In reality the game developer – if simultaneously publishing across several channels – will have to support all price model combinations.

13.3.2 Use case: MGIF Benefit

MGIF requires the game developer to “instrument” their game with the appropriate “event triggers” at potentially chargeable points. The API and data types for these event triggers are an MGIF specification.

When the game “presents” itself to the game platform, it can both explicitly declare what can be charged, and provide the necessary “real time” notification to the game platform over a standard API.

This means that:

- The game developer can move from one platform to another with minimal of rework. (Low porting cost)
- The game can be hosted to two carriers simultaneously using the same base code.
- The service provider/carrier can manage/group the available games by the charging mechanism that they support.
- The service provider/carrier can consistently manage real time pricing for a broad host of games provided by different suppliers.
- The game developer does not have to hard code explicit prices into its game.

Game Platform Provider A and Game Platform Provider B have MGIF compliant platforms that both offer the MGIF metering API to their developers. This is a simple API that requires minimal effort to learn.

When circumstances require a change of game platform, the metering aspects of the games do not have to be reworked, as the API supported by each platform is identical.

13.4. LOGGING

13.4.1 Use Case: Current Status

Game Developer A and Game Developer B develop multiple titles each. As there is no standard for passing logging information from game to game platform, logging tends to be produced for program developer’s eyes only.

This means from the game platform perspective very little information can be gleaned from the running game.

13.4.2 Use Case: MGIF Benefit

MGIF will publish an API to standardise passing of logging information from game to game platform. A service provider for two important functions can use logging information:

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

- To troubleshoot in the case of malfunctioning of badly behaving games
- To gather data for data-warehousing, and data mining purposes.

Logging information that can be made available consistently for a different audience (i.e. the service providers) should improve the quality of services.

Game developers can be assured that info logged out through the standard interface can be transplanted onto another platform with no rework, should encouraging its use.

Logging depth can be increased or decreased in a consistent manner, depending on needs of the service provider.

13.5. SCORE PUBLISHING

13.5.1 Use Case: Current Status

There exists no universal standard for publishing high scores. Game Developer A and Game Developer B each develop multiple titles. Although a proprietary standard may be provided by some games platforms e.g. Game Platform Provider A, it is by no means standard, and Game Platform Provider B may have no equivalent. Consequently not only would it be difficult to port high score functionality; that functionality may not even exist.

Without competition it is hard to stimulate the community spirit that will keep game players coming back for more. Without high scores competitions cannot be run. To ensure publication to the largest possible market, the game developer will have to replicate high score functionality across titles because the game developer cannot guarantee that the functionality is ubiquitous.

13.5.2 Use Case: MGIF Benefit

MGIF will standardize an API to send high scores from a game to a game platform. At the request of the carrier the game platform can then use scores in variety of ways to build community: e.g. simple high score tables, managing one on one matches and seeding players, running large scale knock out competitions.

The benefits are therefore:

- Available functionality. The game developer will not waste time redeveloping competition and score management functionality, as it is guaranteed to be available in an MGIF compliant platform.
- Portability. The game developer can be assured high score functionality can be easily ported from one platform to another because of a standard API.
- Consistency. Integration to a carrier's conventional platform can be at the single integration point (i.e. the game platform) rather than having to reintegrate for every game.
- Interoperability. A carrier/service provider can consistently manage competitions, high scores etc. across a wide range of games provided by different suppliers.

14. MGIF COMPLIANCE

MGIF's goal in a certification procedure is to have the maximum number of games available for the maximum number of platforms by assuring that each platform, game or service conforms to the MGIF specifications. Certification will be achieved, at least in the short term, through developer self-certification. This will be in the form of a checklist that must be completed by the

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.

developer. A series of defined tests to be performed could be used to ensure the developer has adhered to the standards MGIF has set.

Initially MGIF will be concerned with the certification of platforms. Primarily this will check that the platform has correctly implemented the MGIF APIs and that it behaves correctly. Conformant platforms are needed before the certification process can be extended to include games and other services.

If games were certified at this stage it would not extend much past a simple self-certification process. This would be largely to ensure the developer has made use of all relevant MGIF APIs and that they are using them correctly.

15. TERMS AND ABBREVIATIONS

EDGE - Enhanced Data rates for GSM Evolution

EMS - Enhanced Messaging Service

GSM - Global System for Mobile Communication

GPRS - General Packet Radio Services

Mobile Phone - A term often used interchangeably with cellular phone or wireless phone. Initially, a mobile phone referred to a phone attached to a vehicle, the vehicle's battery and had an external antenna. Mobile phones were distinguished from transportable, portable, cordless and personal phones. See also Transportable Phones.http://intromobile.com/glossary/glossary_e.htm

MMS - Multimedia Messaging Service

PDC - Personal Digital Cellular

SMS - Short Message Service

TDMA - Time Division Multiple Access

UMTS - Universal Mobile Telecommunications System

WAP - Wireless Application Protocol

WCDMA - Wideband Code-Division Multiple Access

Wireless - Without wires. Communication without any physical connections between the sender and the receiver. Using the radio frequency spectrum (airways) and hardware, software and technologies to transmit information. http://intromobile.com/glossary/glossary_e.htm

Wireless terminal - Any mobile terminal, mobile station, personal station, or personal terminal using non-fixed access to the network http://intromobile.com/glossary/glossary_e.htm

WLAN - Wireless Local Area Network

XHTML - Extensible Hypertext Markup Language

Copyright © 2002 Mobile Games Interoperability Forum. All rights reserved.