Game Services API Requirements

Approved Version – 11 Sep 2012

Open Mobile Alliance

OMA-RD-GS_API-V1_0-20120911-A

Contents

1. SCOPE (INFORMATIVE) 5
2. REFERENCES 6
2.1 NORMATIVE REFERENCES 6
2.2 INFORMATIVE REFERENCES 6
3. TERMINOLOGY AND CONVENTIONS 7
3.1 CONVENTIONS 7
3.2 DEFINITIONS 7
3.3 ABBREVIATIONS 7
4. INTRODUCTION (INFORMATIVE) 8
4.1 GAME SERVICE API FIGURE 8
5. RELEASE DESCRIPTION (INFORMATIVE) 9
6. REQUIREMENTS (NORMATIVE) 10
6.1 HIGH-LEVEL FUNCTIONAL REQUIREMENTS 10
6.1.1 Security 11
6.1.2 Charging Events 12
6.1.3 Administration and Configuration 12
6.1.4 Usability 12
6.1.5 Interoperability 12
6.1.6 Privacy 12
6.2 OVERALL SYSTEM REQUIREMENTS 13
APPENDIX A. CHANGE HISTORY (INFORMATIVE) 14
A.1 APPROVED VERSION HISTORY 14
APPENDIX B. USE CASES (INFORMATIVE) 15
B.1 GAME RANKING 15
B.1.1 Short Description 15
B.1.2 Market benefits 15
B.2 BUDDY LIST 15
B.2.1 Short Description 15
B.2.2 Market benefit 16
B.3 USER INVENTORY 16
B.3.1 Short Description 16
B.3.2 Market benefit 16
B.4 GAME COMMENT 16
B.4.1 Short Description 16
B.4.2 Market benefit 17
B.5 ITME PURCHASING 17
B.5.1 Short Description 17
B.5.2 Market benefit 17
B.6 SNS LINKAGE 17
B.6.1 Short Description 17
B.6.2 Market benefit 18
B.7 MATCHMAKING 18
B.7.1 Short Description 18
B.7.2 Market benefit 18
B.8 IN-GAME ADVERTISING 18
B.8.1 Short Description 18
B.8.2 Market benefit 18

Figures

No table of figures entries found.

Tables

Table 1: High-Level Functional Requirements **10**

Table 2: High-Level Functional Requirements – Security Items **11**

Table 3: High-Level Functional Requirements – Authentication Items **11**

Table 4: High-Level Functional Requirements – Authorization Items **11**

Table 5: High-Level Functional Requirements – Data Integrity Items **11**

Table 6: High-Level Functional Requirements – Confidentiality Items **11**

Table 7: High-Level Functional Requirements – Charging Events Items **12**

Table 8: High-Level Functional Requirements – Administration and Configuration Items **12**

Table 9: High-Level Functional Requirements – Usability Items **12**

Table 10: High-Level Functional Requirements – Interoperability Items **12**

Table 11: High-Level Functional Requirements – Privacy Items **13**

Table 12: High-Level System Requirements **13**

1. Scope (Informative)

This document presents the Requirements for the OMA Game Service API 1.0, and the Use Cases from which they are derived. The Requirements herein should be understood as those minimally necessary to implement a service that supports connect and multiplayer gaming over mobile networks and device.

The Requirements fall in to five distinct phases of the User's interaction with the Game Service API: obtaining the game ranking, in-game-SNS, item purchasing, match making, in-game-advertising.

2. References

2.1 Normative References

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

2.2 Informative References

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_8, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

User (also: end-user, player, game player, gamer, customer, consumer)	The human being who is playing a game.
Device (also: handset, game device, mobile device, phone, mobile phone)	The electronic terminal on which the user is playing a game. It is assumed to be able to send and receive information via a mobile operator’s network.
Game Service	The collection of end-to-end functionality provided that enables connected and/or multiplayer mobile games.
Game Service Provider	The business entity that provides the functionality of a Game Service to Users. A GSP may provide and manage all Game Service functionality itself, or make these available through business-to-business relationships.
Game Client	The portion of a mobile game (connected, multiplayer) that executes on the User’s device that is specifically concerned with interacting with the OMA Game Service. General reference to the game client excludes device-resident game fiction.
Mobile application (game)	An application that executes on a mobile device. (A mobile application that is a game).
Connected (multiplayer) mobile game	A mobile game in which some of the User’s experience involves entities that are not resident on the User’s device; a Lottery is a connected mobile game, as is a game that admits high-score posting, or the downloading of ghost racers. (A mobile game in which multiple players simultaneously interact with one another through, or as a necessary part of, the game fiction).
Turn-based Game	A game in which, at each point in time, only a subset of players are allowed to make a move that alters the game’s shared state.
Matchmaking	The process whereby a User indicates interest in participating in a multiplayer mobile application (multiplayer game), possibly specifying criteria that the other participants must meet. The result is either a game session in which the User has been placed, or an indication that it is not possible to meet the specified criteria.

3.3 Abbreviations

OMA	Open Mobile Alliance
GS API	Game Service API

4. Introduction (Informative)

The mobile game service market has changed and the current trend of mobile game service is about smart-phones which can get a lot of mobile games through open App-market. GS API is the standard to develop server-client network function for mobile game.

Game developer can develop some network functions like game score ranking, item purchasing, posting game score in easy way by using GSAPI. GSAPI will stimulate mobile game market by providing easy way to develop network functions for game developers. And also GS API can help developers to make better mobile games in short time.

Operators

App store operators can gather better games and expect more revenue by in-game-charging commission. In-game-SNS will stimulate the game users to download games more to enjoy with their friends.

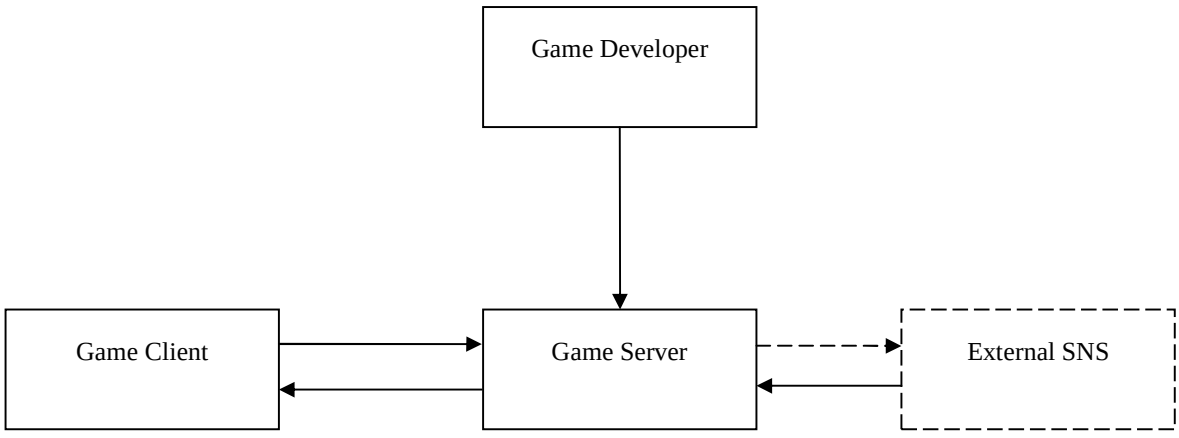
3rd Party developers

Game developers only have to use GS API service to make network functions for game so that they can make game content i n short time. Moreover, because they can concentrate only in client game programming, they can develop qualified games. 3rd Party developers can get more revenue by in-game-charging like operators.

Users

Game users can get more good games and enjoy the games more with their friends by in game SNS.

4.1 Game Service API figure



5. Release Description (Informative)

5.1 Version 1.0

This is the first version of GS API RD which specifies the Game service API requirements and describes some usecases.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

Label	Description	Release
GSAPI - HLF-001	The GSAPI MAY be capable to provide Game user's game score to other users. Game user can see other player's game score. The score data should be saved in game server and the game client can get the data when the game users want by calling API.	GSAPI V1.0
GSAPI - HLF-002	The GSAPI MAY be capable to provide the function for Game user's social network information. Game users can see the buddy who enjoy same game and can see the game list of the buddy. Game users can add the buddy to their buddy list. It should be saved in game server.	GSAPI V1.0
GSAPI - HLF-003	The GSAPI MAY be capable to provide the function to buy some game items in playing game. Game developers can add their game item to game server database and if game users buy that game item, the game server notify that to the game client.	GSAPI V1.0
GSAPI - HLF-004	The GSAPI MAY be capable to provide the function for connect with other player to enjoy same game simultaneously.	GSAPI V1.0
GSAPI - HLF-005	The GSAPI MAY be capable to provide the function for in-game-advertisement.	GSAPI V1.0
GSAPI –HLF-006	GSAPI SHALL be able to provide the function for game users to send message to their buddy.	GSAPI V1.0
GSAPI –HLF-007	GSAPI SHALL be able to provide the function for game users to manage their game items which were bought. Game users can see the buying history or item status in inventory.	GSAPI V1.0
GSAPI –HLF-008	GSAPI SHALL be able to provide the function for game users to give some comment about a game which they enjoyed. Game users can see other player's comment about the game.	GSAPI V1.0
GSAPI –HLF-009	GSAPI SHALL be able to provide the function for game users to post their game score or game comment to other SNS service.	GSAPI V1.0
GSAPI –HLF-010	GSAPI SHALL be able to provide the function for game users to register their account(ID and password, email) to the game server.	GSAPI V1.0
GSAPI –HLF-011	GSAPI SHALL be able to provide the function for game users to find buddies whom the game user wants to add to his/her buddy list. Buddies can be found by nickname or CID.	GSAPI V1.0
GSAPI –HLF-012	GSAPI SHALL be able to provide the function for game developers to register his/her game to the GSAPI service through game developer's portal.	GSAPI V1.0
GSAPI –HLF-013	GSAPI SHALL be able to provide the function for game developers to register the game item which will be used in a game.	GSAPI V1.0
GSAPI –HLF-014	GSAPI SHALL be able to provide the function for game developers to register his/her information to the GSAPI server to provide GSAPI service to the game user. Developers can register their information through the developer's portal.	GSAPI V1.0

Table 1: High-Level Functional Requirements

6.1.1 Security

Label	Description	Release
GSAPI-SEC-001	The GSAPI Enabler SHALL support secure delivery of game data.	GSAPI V1.0
GSAPI -SEC-002	The GSAPI Enabler SHALL support various security mechanisms for protect game user's information.	GSAPI V1.0
GSAPI -SEC-003	The GSAPI Enabler SHALL be able to use keys needed to support confidentiality, integrity protection, and authenticity.	GSAPI V1.0
GSAPI -SEC-004	The GSAPI Enabler SHALL support secure storage of data (e.g., validated user properties).	GSAPI V1.0

Table 2: High-Level Functional Requirements – Security Items

6.1.1.1 Authentication

Label	Description	Release
GSAPI-AUC-001	The GSAPI Enabler MUST be able to support the authentication of game Users.	GSAPI V1.0
GSAPI -AUC-002	The GSAPI Enabler MAY support validation of certificates for certain use cases, such as the attestation of personal user information.	GSAPI V1.0

Table 3: High-Level Functional Requirements – Authentication Items

6.1.1.2 Authorization

Label	Description	Release
GSAPI -AUZ-001	The GSAPI enabler SHALL be able to authorize the game Users.	GSAPI V1.0

Table 4: High-Level Functional Requirements – Authorization Items

6.1.1.3 Data Integrity

Label	Description	Release
GSAPI -DIT-001	The GSAPI Enabler SHALL support integrity of data.	GSAPI V1.0
GSAPI -DIT-002	The GSAPI Enabler SHALL support data integrity in protecting against accidental or intentional changes to game-related data transmission, by ensuring that changes to the data are detectable.	GSAPI V1.0

Table 5: High-Level Functional Requirements – Data Integrity Items

6.1.1.4 Confidentiality

Label	Description	Release
GSAPI-CON-001	The GSAPI Enabler SHALL support encryption of messages.	GSAPI V1.0
GSAPI -CON-002	The GSAPI Enabler SHALL support decryption of messages.	GSAPI V1.0
GSAPI -CON-003	The GSAPI Enabler SHALL support data confidentiality that ensures transmitted information is not made available to unauthorised individuals or entities.	GSAPI V1.0

Table 6: High-Level Functional Requirements – Confidentiality Items

6.1.2 Charging Events

Label	Description	Release
GSAPI-CHG-001	The GSAPI Enabler SHALL support means to charge for buying game item	GSAPI 1.0
GSAPI -CHG-002	The charging MUST use standardized mechanisms.	GSAPI 1.0
GSAPI -CHG-003	The GSAPI Enabler SHALL enable the charging for delivery of information to third parties.	GSAPI 1.0

Table 7: High-Level Functional Requirements – Charging Events Items

6.1.3 Administration and Configuration

Label	Description	Release
GSAPI- ADM-001	The game users SHALL have the possibility of multiple subscriptions with different Service providers.	GSAPI 1.0
GSAPI -ADM-003	The GSAPI Enabler SHOULD support collection of usage data from the Game Users.	GSAPI 1.0

Table 8: High-Level Functional Requirements – Administration and Configuration Items

6.1.4 Usability

Label	Description	Release
GSAPI-USE-001	GSAPI SHALL be able to provide appropriate usability.	GSAPI 1.0

Table 9: High-Level Functional Requirements – Usability Items

6.1.5 Interoperability

Label	Description	Release
GSAPI- INT-001	The GSAPI Enabler SHALL allow game users to access any Service (i.e. user-managed or operator-managed Services as well as 3rd party Services) on any suitable device (i.e. a device matching the capabilities required to consume the Service) and within any network island (e.g. home, car, hotspot, hotel, friend's place or office).	GSAPI 1.0
GSAPI- INT-002	The GSAPI Enabler SHALL be able to interoperate with game user's electronic devices.	GSAPI 1.0

Table 10: High-Level Functional Requirements – Interoperability Items

6.1.6 Privacy

Label	Description	Release
GSAPI -PRIV-001	The GSAPI Enabler SHALL ensure user privacy.	GSAPI 1.0
GSAPI -PRIV-002	The privacy requirements in [OMA-Privacy] SHALL be applied to the GSAPI Enabler.	GSAPI 1.0
GSAPI -PRIV-003	The GSAPI User SHALL be able to configure privacy policies for the management of game data .	GSAPI 1.0
GSAPI -PRIV-004	The GSAPI Enabler SHALL support the game Users to verify whether they accept the collection of their usage data.	GSAPI 1.0

GSAPI -PRIV-005	The GSAPI Enabler SHALL support the game Users to cancel the collection of their usage data.	GSAPI 1.0
GSAPI -PRIV-006	The GSAPI Enabler SHALL be able to protect personal user information when transmitting and performing a test on it.	GSAPI 1.0

Table 11: High-Level Functional Requirements – Privacy Items

6.2 Overall System Requirements

Label	Description	Release
GSAPI-SYS-001	It SHALL be possible for GSAPI Enabler to be deployed in game server.	GSAPI 1.0
GSAPI S-SYS-002	The GSAPI Enabler SHOULD support reporting of statistics to authorized receiving parties.	GSAPI 1.0

Table 12: High-Level System Requirements

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
Approved version: OMA-RD-GS_API-V1_0	11 Sep 2012	Status changed to Approved by TP: TP ref#: OMA-TP-2012-0333-INP_GS_API_V1_0_RRP_for_Final_Approval

Appendix B. Use Cases (Informative)

B.1 GAME RANKING

Game ranking function supports game users to save their own game score in the game server and to see the game scores of themselves and of other game players.

B.1.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

The game ends and game user can see the game score.
Game user clicks the button to save the score.

Game client sends the score to the server.

Game server saves the score to database.

If a game player wants to see the all game score, he clicks ranking button. Game client get the scores by using game service API. The player can see the scores and game ranking

B.1.2 Market benefits

Game user can see the game ranking and get more desire to achieve better score.
Game developers can add the game ranking function in their own game in easy and quick way by using API.

B.2 BUDDY LIST

Buddy list function supports game users to find and add a buddy in the game server and to retrieve previously added their buddy list.

B.2.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user finds his or her buddy by e-mail or nickname and adds the buddy to his or her list.

Game client sends a request for adding the buddy to the server.

Game server saves the buddy to database.

If a game player wants to see the buddy, he clicks buddy button. Game client gets his buddy list by using game service API and the player can see the buddy list and recursively buddy's buddy list also.

B.2.2 Market benefit

Game users can see their buddy list and recursively buddy's buddy list so it can be easy to add new buddy by using this function.
Game developers can add player's buddy list function in their own game in easy and quick way by using API.

B.3 USER INVENTORY

User inventory function supports game users to get their inventory previously purchased by themselves.

B.3.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

Game user gets his or her inventory.

Game client sends a request to get inventory.

Game server retrieves his or her inventory at database.

If a game player wants to see the list of inventory, he or she clicks the inventory button. Game client get the list by using game service API.

B.3.2 Market benefit

Game users can see their item inventory list and purchase history by using this function
Game developers can add player's inventory list function in their own game in easy and quick way by using API.

B.4 GAME COMMENT

Game comment function supports game users to save a comment about the game and to get comment list of the game.

B.4.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

Game user adds a new comment about the game.

Game client sends a request to add the comment.

Game server saves the comment to database.

If game user wants to get comment list of the game, he or she clicks the game comments button. Game client get the list of comments by using game service API.

B.4.2 Market benefit

Game users can add their comments about a game by using this function.
Game developers can add game comment function in their own game in easy and quick way by using API.

B.5 ITME PURCHASING

Item purchasing function supports game users to purchase a new item in a game.

B.5.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

Game user finds attractive new item and decides to purchase the item.

Game client sends a request to purchasing the item.

Game server processes to purchase the item.

If the processing for purchasing the item succeeds, game client get the code for unlocking the item by using game service API.

B.5.2 Market benefit

Game users can purchase a new item by clicking a buying button.
Game developers can add item purchase function in their own game in easy and quick way by using API.
Game developers can expect more revenue by selling game items.

B.6 SNS LINKAGE

SNS linkage function supports game users to publish a comment related to the game.

B.6.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

Game user gets a new personal record of the game and wants to share the game events with his or her friends of SNS.

Game client sends a request to publish the game events.

Game server publishes the events to SNS based on the game player's SNS information.

If a game player wants to publish a new record of a game, he clicks posting button. Game client sends a request to server by using game service API.

B.6.2 Market benefit

Game users can share some game events with friends of their SNS.
Game developers can add SNS linkage function in their own game in easy and quick way by using API.

B.7 MATCHMAKING

Matchmaking function supports game users to find other players for multi play in the game.

B.7.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game supporting multi play.

Game user clicks the ONLINE or MULTI-PLAY button.

Game client sends a request to find an appropriate player.

Game server matchmakes players.

If a game player wants to play online, he clicks online or multiplay button. Game client send a request to server by using game service API.

B.7.2 Market benefit

Game users can play with appropriate players.
Game developers can add matchmaking function in their own game in easy and quick way by using API.

B.8 IN-GAME ADVERTISING

In-Game advertising function supports game developers to add an advertisement in their game.

B.8.1 Short Description

- Game user
- Game developer
- Game server
- Game client

Game user plays a game.

Game user clicks the advertisement banner in a game.

Game client sends a request to game server.

Game server saves the click history to database.

If a game developer wants to add advertisement, he adds an advertisement by using game service API.

B.8.2 Market benefit

Game developers can add in-game advertising function in their own game in easy and quick way by using API.