OMA Game Services

Client/Server Interface Specification

Candidate Version 1.0 – 07 Mar 2006

Open Mobile Alliance

OMA-TS-Game-Services-Client-Server-Interface-V1_0-20060307-C

Contents

Figures

Tables

1. Scope

The Open Mobile Alliance (OMA) Game Services (GS) Client-Server Interface (CSI) is a result of continuous work to define an industry-wide specification for developing game applications that operate over wireless communication networks.

The scope for the Open Mobile Alliance is to define a set of specifications to be used by service applications. The wireless market is growing very quickly, and reaching new customers and providing new services. To enable operators, service providers and manufacturers to meet the challenges in advanced services, differentiation, and fast/flexible service creation, the OMA-GS CSI enabler defines a set of interfaces in (transport, session and) application layers. For additional information on the OMA-GS CSI architecture, refer to "Game Services Architecture" [GSAD].

The scope of OMA-GS "Client Server Interface" is to enable the controlled usability of wireless games, e.g. the ability to execute games and have a standardized interaction with the game server. The defined technology is an initial Game Service System that can be extended into a more comprehensive Game Service System in the future.

2. References

2.1 Normative References

[GP1.0] "Gaming Platform", Version 1.0, Open Mobile Alliance™, OMA-GamingPlatform-V1_0, URL:http://www.openmobilealliance.org/

[GSAD] "Game Services Architecture", Version 1.0, Open Mobile Alliance™, OMA-Gaming Architecture-Overview-V1_0, URL:http://www.openmobilealliance.org/

[HTTP] "Hypertext Transfer Protocol", IETF, RFC 2616, URL:http://www.ietf.org

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.3, Open Mobile Alliance™, OMA-IOP-Process-V1_3, URL:http://www.openmobilealliance.org/

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

[RFC2234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt

[SMS] "The technical Realization of SMS". 3GPP TS 23.040-v.5.3.0, URL:http://www.3gpp.org

[SOAP] "Simple Object Access Protocol", Version 1.1, W3C,

URL:http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[TCP] "Transmission Control Protocol – DARPA Internet Program Protocol Specification", IETF, RFC 793, URL:http://www.ietf.org

[UAPROF] http://www.openmobilealliance.org/tech/profiles/index.html

[UDP] "User Datagram Protocol", IETF, RFC 768, URL:http://www.ietf.org

[WSDL] "Web Services Description Language(WSDL)", Version 1.1, W3C,

URL:http://www.w3.org/TR/wsdl

[XML] "eXensible Markeup Language", W3C, URL:http://www.w3.org/XML/

2.2 Informative References

[MIDP1.0] "Mobile Information Device Profile for the J2ME Platform Version 1.0",JSR-37, Java Community Process, URL:http://www.jcp.org/

[MIDP2.0] "Mobile Information Device Profile Version 2.0",JSR-118, Java Community Process, URL:http://www.jcp.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

3.3 Abbreviations

AI	Application Instance
AppID	Application ID
API	Application Programming Interface
AS	Actor Session
ASID	Actor Session ID
CSI	Client Server Interface
DBG	DeBuG message
ERR	ERRor code
GS	Game Services
HTTP	Hyper Text Transfer Protocol
HTTPS	Secure Hyper Text Transfer Protocol
ID	IDentity
INT	Integer
JSR	Java Specification Request
MIDP	Mobile Information Device Profile
MMS	Multimedia Message Service
MSG	MeSsaGe
OMA	Open Mobile Alliance
P2P	Peer TO Peer
POS	POSition
SID	Session ID
SCR	Statistic Conformance Requirements

SMS	Short Message Service
SOAP	Simple Object Access Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

4. Introduction

This specification defines the interface between the client (e.g. mobile phone) and the server. Client and Server are considered as black boxes.

The main part of the interface specification is the *Client/Server Contract* which defines the communication between the Client and the Server. To implement the interface it is important to understand the underlying domain model and concepts which are specified as well. An overview is shown in the following figure:

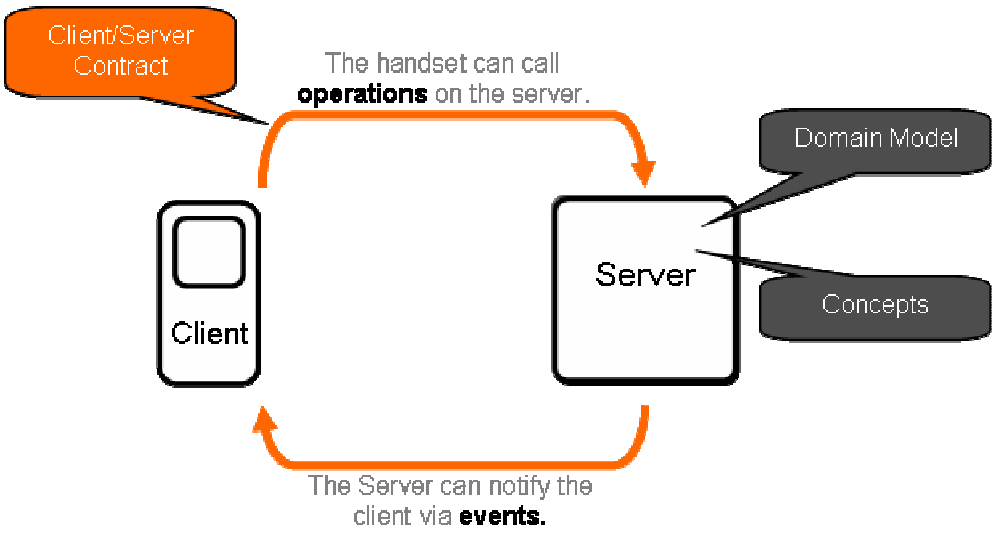

Figure 1: Solution overview

The *Concepts* are specified in Chapter 5 (Basic Concepts).

The server-side *Domain Model* is specified in Chapter 5.2 (Domain Model).

The *Client/Server Contract* is specified in Chapter 6 (Client/Server Interface Contract).

The main design goals, for this Client/Server Interface specification, are:

1) Programming language independence:
   The interface can be implemented using any of a variety of programming languages.

2) Protocol independence:
   The client/server interface defined in this specification can be implemented using a variety of protocol-technologies, e.g., web–services [WSDL], SMS [SMS], HTTP [HTTP], TCP [TCP] or UDP [UDP].

5. Basic Concepts

5.1 Types of Games

There are different basic types of games, as described in the following table (combinations are feasible):

CSI 1.0	#	Game Type	Examples	Characteristics
	1	**Soloplay**		
✓	1.1	Soloplay Highscore Game	- Shoot'em-up games - Platform games - Puzzle games	- 1 Player - no need to hit the network - communication not critical to gameplay
✓	*1.1.1*	*Arcade*		- *Multiple highscore entries per player*
✓	*1.1.2*	*Ranking Single / Championship*		- *One highscore (rank) per player*
✓	*1.1.3*	*Ranking Sum*		- *One highscore (rank) per player. Highscore is added up*
✓	1.2	Soloplay Shadow Game	- "Shadow Gaming" - Racing games	- 1 Player - need to hit the network - feels like a multiplayer game, but is a single player challenge
✓	*1.2.1*	*Arcade*		- *Multiple highscore entries per player*
✓	*1.2.2*	*Ranking Single / Championship*		- *One highscore (rank) per player*
	2	**Turn based**		
✓	2. 1	Round Robin	- Board Games (e.g. Chess, Backgammon, etc.) - Strategy games	- 2 – n players - Each player makes turn (defined order) - Next Round after last player in order finished
✓	2. 2	Simultaneous Movement	- Strategy games	- 2 – n players - Every player makes turn (parallel) - Next round after every player finished and all turn data was sent to all players simultaneously
✓	2. 3	Custom Turn	- Strategy games	- 2 – n players - Custom turn - decided by the game client

	3	**Act Whenever**		
	3.1	Act Whenever	- "Multi User Dungeons" - Virtual Worlds	- 1 – n players - User's representation within the game (e.g. character) is unavailable when the user is offline
	4	**Slow update**		
	4.1	Slow update	- Long term strategy games - Virtual Pets	- 1 – n players - User's representation within the game (e.g. character) stays available, executes commands or acts independently even when the user is offline
	5	**Real Time**		
	5.1	Real Time	- Racing games - 3D Shooter	- 2 – n players - Users are connected directly and can compete against each other without latency time

Table 1: Different type of games

The focus of this specification will be "Soloplay" games (type 1), except ‘Soloplay Highscore Game’, which has characteristics of ‘no need to hit the network’, and "Turn Based" games (type 2).

Further game types might be supported with extended interfaces in future versions of Game Services specifications.

5.2 Domain Model

The domain model residing in the game server can be seen in Figure 2:

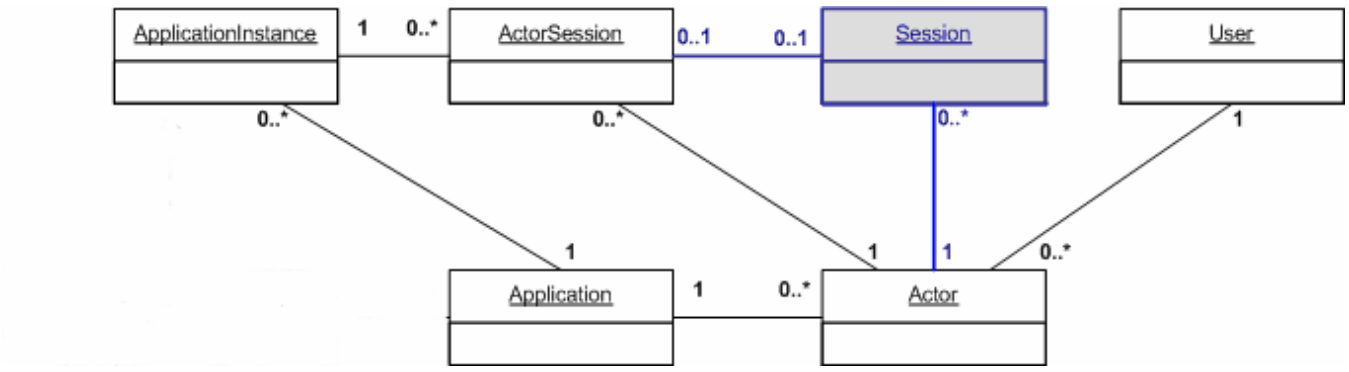

Figure 2: The server side Domain model

This model is an evolution of the OMA GP 1.0 domain model [GP1.0]. A new "Session" entity has been introduced in this specification. Table 2 describes the domain model entities (with examples):

Entity	Description	Example
User	Unique identification of one User (or Player) by an account consisting of a username and a password.	Dave Smiths' *User* is identified by Username "Dave1000" and Password "hello!13"
Actor	The platform internal representation of one *User* for one *Application.* A *User* has one *Actor* for each *Application.*	Dave Smith's *User* has one *Actor* for playing Chess and one for playing Backgammon
Session	The Session identifies the Actor during the communication between client and server. One Actor can have multiple Sessions.	If Dave logs on for Chess a new *Session* for his *Actor* is generated.
Application	The actual application – in this case, most likely a game	Chess, Backgammon, etc.
ApplicationInstance	Representation of a running game of the corresponding *Application* and the connected *Actors.* Every *Application* can have multiple *ApplicationInstances*	Dave Smith's *Actor* for Chess is connected to two *ApplicationInstances*: "Chess match between Dave Smith and Peter Miller" and "Chess match between Dave Smith and John Doe"

ActorSession	The *ActorSession* is the link between an *Actor* and the *ApplicationInstance.* For every *ActorSession* there is one *ApplicationInstance.* An Actor can join multiple *ApplicationInstances* via multiple *ActorSessions*. Only one *ActorSession* per *Actor* can be active – the *Session* points to it.	Two *ActorSessions* are connected to Dave's *Actor.* One pointing to the *ApplicationInstance* of "Chess match between Dave Smith and Peter Miller" and the other *ActorSession* pointing to the *ApplicationInstance* "Chess match between Dave Smith and John Doe"

Table 2: Domain Model entities description with examples

6. Client/Server Interface Contract

The interface is defined by a Web Service (WSDL [WSDL]) contract, which consists of Operations and Events. Datatypes are defined in XML Schema [XML]:

6.1 Introduction

Each section that describes an operation includes the following items:

- A table including all parameters used when calling this operation and all return values that are being sent back by the operation to the client. This approach was used to get an easy to read, yet complete description of each operation.
- An XML Schema [XML] definition of the request and response types used for the particular operation. They are part of the WSDL [WSDL] document describing the whole Game Platform Service. The main point to realize is that the description of the request and response types are equivalent to the enumeration of parameter and return values (along with datatypes) shown in the aforementioned table. The XML Schemas are more concise and thus initially more difficult to read, because they make use of inheritance (for including frequently used parameters and return values) and references (this relates to datatype definitions) which basically means this information needs to be looked up elsewhere in the document. The main advantage is that they are machine-readable and indicate what class in terms of common parameters and return values this operation belongs to.
- A sequence diagram illustrating the message flow between client and server. All items depicted to the right of the SOAP [SOAP] interface belong to the server. As such, they are not part of the client/server interface contract, and are included merely for illustration purposes of how the server part might be implemented.
- To improve readability, the sequence diagrams will not show the standard return values (SID, ASID, DBG, ERR).

Below you find examples for the table and XML Schema definitions.

<table>
<tr><td>Name</td><td>OperationName(Param1, Param2, …)</td></tr>
<tr><td>Description</td><td>Description of the Operation. Domain Model Entities are boldfaced: DomainModelEntity</td></tr>
<tr><td>Parameter</td><td>
<table>
<tr><th>Shortcut</th><th>Data type</th><th>Value</th></tr>
<tr><td>Param</td><td>…</td><td>Description.</td></tr>
</table>
</td></tr>
<tr><td>Return Values</td><td>
<table>
<tr><th>Shortcut</th><th>Data type</th><th>Value</th></tr>
<tr><td>RetVal</td><td>…</td><td>Description</td></tr>
</table>
Default return values for all operations:
<table>
<tr><th>Shortcut</th><th>Data type</th><th>Value</th></tr>
<tr><td>ERR</td><td>Integer</td><td>Error Code
0 (zero) : OK
Otherwise : Error</td></tr>
<tr><td>DBG (optional)</td><td>String</td><td>Debug Message</td></tr>
<tr><td>SID (optional)</td><td>String</td><td>Session ID</td></tr>
</table>
</td></tr>
<tr><td>Related operations/events</td><td>Link to related operations.</td></tr>
<tr><td>Errors</td><td>Possible Errors.</td></tr>
</table>

```
<xsd:complexType name="OperationRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:BaseCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Param1" />
         ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="OperationResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:BaseCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:RetVal1" />
        ...
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.1.1 Common parameters and return values

There are four classes of operations. All operations belonging to one class share the same set of common parameters (request) and return values (response). These classes build up on each other and thus form a class hierarchy. An operation belongs to a certain class of operations and inherits their common parameters and return values. They are described here.

In the WSDL definitions of the operations, common parameter and return values inherited from the operation class are not listed again, but only a reference to the operation class. The parameters and return values are additionally defined in XML Schema (XSD). The definitions for the common parameters and return values are:

```
<xsd:complexType name="DefaultCommandRequest">
</xsd:complexType>
```

```
<xsd:complexType name="SessionCommandRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:SID" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="ASCommandRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ASID" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="ActiveASCommandRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="DefaultCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ERR" />
        <xsd:element ref="tns:DBG" minOccurs="0" />
      </xsd:sequence>
</xsd:complexType>
```

```
<xsd:complexType name="SessionCommandResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:SID" minOccurs="0" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="ASCommandResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="ActiveASCommandResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.1.2 Simple datatypes

The WSDL definitions of the operations contain operations specific types, that map certain element names to standard dataypes defined in the XMLSchema standard namespace. They are the following:

```
<xsd:element name="ERR" type="xsd:integer" />
<xsd:element name="DBG" type="xsd:string" />
<xsd:element name="SID" type="xsd:string" />
<xsd:element name="ASID" type="xsd:string" />
```

(Parameters and return values for request / response type classes)

```
<xsd:element name="AppID" type="xsd:string" />
<xsd:element name="Username" type="xsd:string" />
<xsd:element name="Password" type="xsd:string" />
<xsd:element name="NumAS" type="xsd:integer" />
<xsd:element name="NumINV" type="xsd:integer" />
<xsd:element name="ActorSessions" type="tns:vector" />
<xsd:element name="Name" type="xsd:string" />
<xsd:element name="Actors" type="xsd:string" />
<xsd:element name="MinAS" type="xsd:integer" />
<xsd:element name="MaxAS" type="xsd:integer" />
<xsd:element name="DTCreated" type="xsd:dateTime" />
<xsd:element name="State" type="xsd:integer" />
<xsd:element name="Active" type="xsd:boolean" />
<xsd:element name="NumInQueue" type="xsd:integer" />
<xsd:element name="ActorNr" type="xsd:integer" />
<xsd:element name="NumInGame" type="xsd:integer" />
<xsd:element name="NumActive" type="xsd:integer" />
<xsd:element name="Data" type="xsd:anyType" />
<xsd:element name="Msg" type="xsd:string" />
<xsd:element name="ScoreTable" type="xsd:string" />
<xsd:element name="Score" type="xsd:long" />
<xsd:element name="ScoreID" type="xsd:string" />
<xsd:element name="TimeFrame" type="xsd:string" />
<xsd:element name="Pos" type="xsd:integer" />
<xsd:element name="Count" type="xsd:integer" />
<xsd:element name="Filter" type="xsd:integer" />
<xsd:element name="Scores" type="tns:vector" />
<xsd:element name="Country" type="xsd:string" />
<xsd:element name="Offset" type="xsd:integer" />
<xsd:element name="Shadow" type="xsd:base64Binary" />
<xsd:element name="TotalBytes" type="xsd:integer" />
```

(Parameters and return values for operations)

6.1.3 Composite datatypes

The WSDL definitions of the operations contain a few composite types that are not defined in the XMLSchema standard namespace. They are the following:

```
  <!-- subitem for Event -->
  <xsd:complexType name="item">
    <xsd:sequence>
      <xsd:element name="key" type="xsd:anyType" />
      <xsd:element name="value" type="xsd:anyType" />
    </xsd:sequence>
  </xsd:complexType>

  <!-- subitem for Event -->
  <xsd:complexType name="codeitem">
    <xsd:sequence>
      <xsd:element name="key" type="xsd:string" />
      <xsd:element name="value" type="xsd:string" />
    </xsd:sequence>
  </xsd:complexType>

  <!-- Single event -->
  <xsd:complexType name="Event">
    <xsd:sequence>
      <xsd:element name="code" type="tns:codeitem" minOccurs="1" maxOccurs="1" />
      <xsd:element name="entries" type="tns:item" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>

  <!-- Unbounded list with items of type Event -->
  <xsd:complexType name="Events">
    <xsd:sequence>
      <xsd:element ref="tns:Event" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>

 <!-- Single ActorSession information -->
  <xsd:complexType name="ActorSession">
    <xsd:sequence>
      <xsd:element ref="tns:ASID" />
      <xsd:element ref="tns:Name" />
      <xsd:element ref="tns:Actors" />
      <xsd:element ref="tns:MinAS" />
      <xsd:element ref="tns:MaxAS" />
      <xsd:element ref="tns:DTCreated" />
    </xsd:sequence>
  </xsd:complexType>

  <!-- Unbounded list with items of type ActorSession -->
  <xsd:complexType name="ActorSessions">
    <xsd:sequence>
      <xsd:element ref="tns:ActorSessionItem" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
```

```
<!-- Single score item -->
<xsd:complexType name="Score">
  <xsd:sequence>
    <xsd:element ref="tns:ScoreID" />
    <xsd:element ref="tns:Pos" />
    <xsd:element ref="tns:Username" />
    <xsd:element ref="tns:Country" />
    <xsd:element ref="tns:Score" />
    <xsd:element ref="tns:DTCreated" />
  </xsd:sequence>
</xsd:complexType>

<!-- Unbounded list with items of type Score -->
<xsd:complexType name="Scores">
  <xsd:sequence>
    <xsd:element ref="tns:Score" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>

<!-- Single Actor -->
<xsd:complexType name="Actor">
  <xsd:sequence>
    <xsd:element ref="tns:ActorNr" />
    <xsd:element ref="tns:Username" />
    <xsd:element ref="tns:State" />
  </xsd:sequence>
</xsd:complexType>

<!-- Unbounded list with items of type Actor -->
<xsd:complexType name="Actors">
  <xsd:sequence>
    <xsd:element ref="tns:Actor" minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

6.2 Operations (called from the client)

6.2.1 Login and Registration

- Login
- Logout
- Register

6.2.1.1 Login

Name	**Login**(AppID, Username, Password)
Description	Logs in a ***User*** for a specific ***Application***. Or described simpler: Logs in an ***Actor***. A new ***Session*** is created and a Session ID (SID) is returned. This session is the key for all other operation calls. The number of assigned ***ActorSessions*** and Invitations is returned to reduce roundtrips to the server.
Parameter	**Shortcut** / **Datatype** / **Value** **AppID** / String / Unique Application ID (provided by the platform). **Username** / String / Valid[1]username. **Password** / String / Valid password.
Return Values	**Key** / **Datatype** / **Value** **ERR** / Int / Errorcode (0 if no error occurred) **DBG** / String / Error message (optional) **NumAS** / Int / Number of ActorSessions currently assigned to the Actor **NumINV** / Int / Number of ActorSessions currently assigned to the Actor
Related operations/events	GetAS, GetInv, EvStatus
Errors	USER_NOT_REGISTERED, USER_WRONG_PASSWORD

6.2.1.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="LoginRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:AppId" />
        <xsd:element ref="tns:Username" />
        <xsd:element ref="tns:Password" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="LoginResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:NumINV" />
        <xsd:element ref="tns:NumAS" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.1.3 Sequence diagram

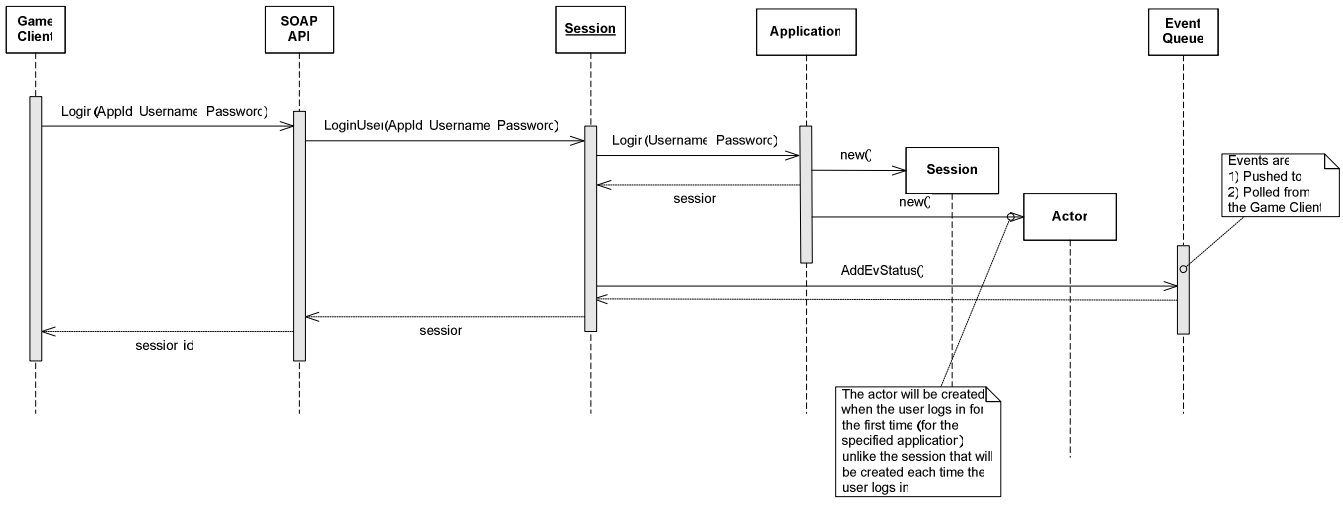

Figure 3: Login Operation (App User)

6.2.1.2 Logout

Name	**Logout**(SID)
Description	Logs out an ***Actor***. The ***Session*** (SID) is invalidated.
Parameter	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**SID**</td><td>String</td><td>Unique Subscription id (provided by the platform).</td></tr></table>
Return Values	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**ERR**</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr><tr><td>**DBG**</td><td>String</td><td>Error message (optional)</td></tr><tr><td>**SID**</td><td>String</td><td>The now invalid session id</td></tr></table>
Related operations/events	Login, EvStatus
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.1.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="LogoutRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="LogoutResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.2.3 Sequence diagram

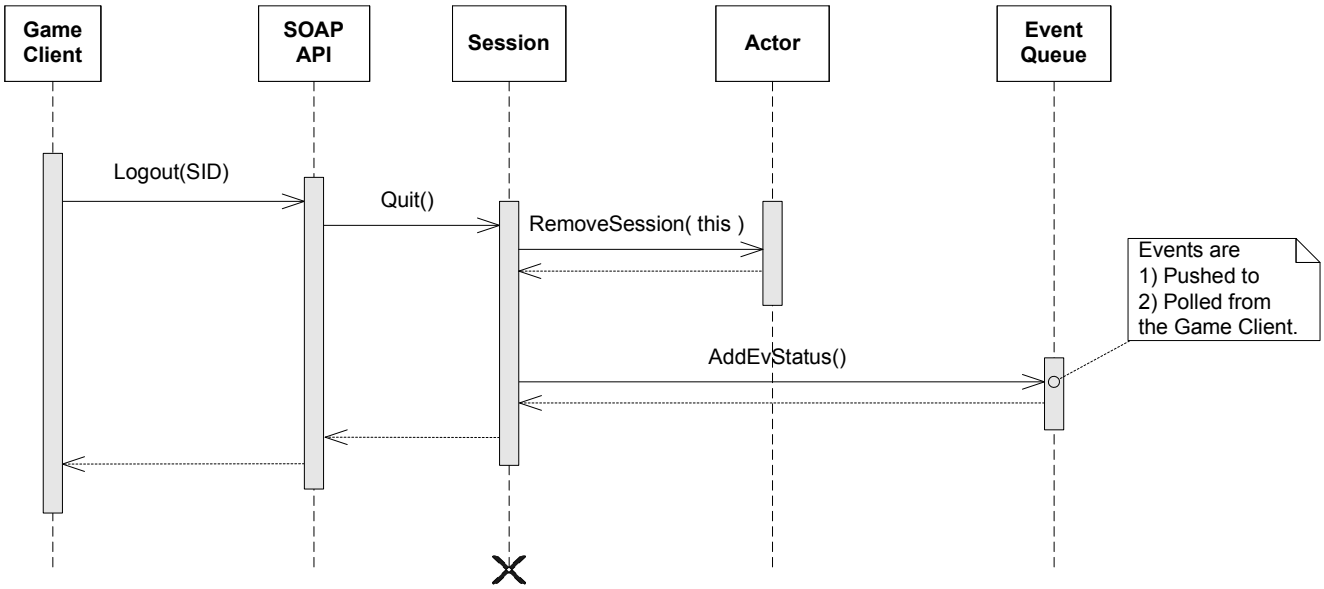

Figure 4: Logout Operation

6.2.1.3 Register

Name	**Register**(SID, Username, Password)
Description	Registers a new ***User***. This ***User*** can use any ***Application*** on the server (if he has the appropriate rights).
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
Username	String	Valid username.
Password	String	Valid password.

Return Values	

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)

Related operations/events	Login
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.1.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="RegisterRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="RegisterResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:DefaultCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.1.3.3 Sequence diagram

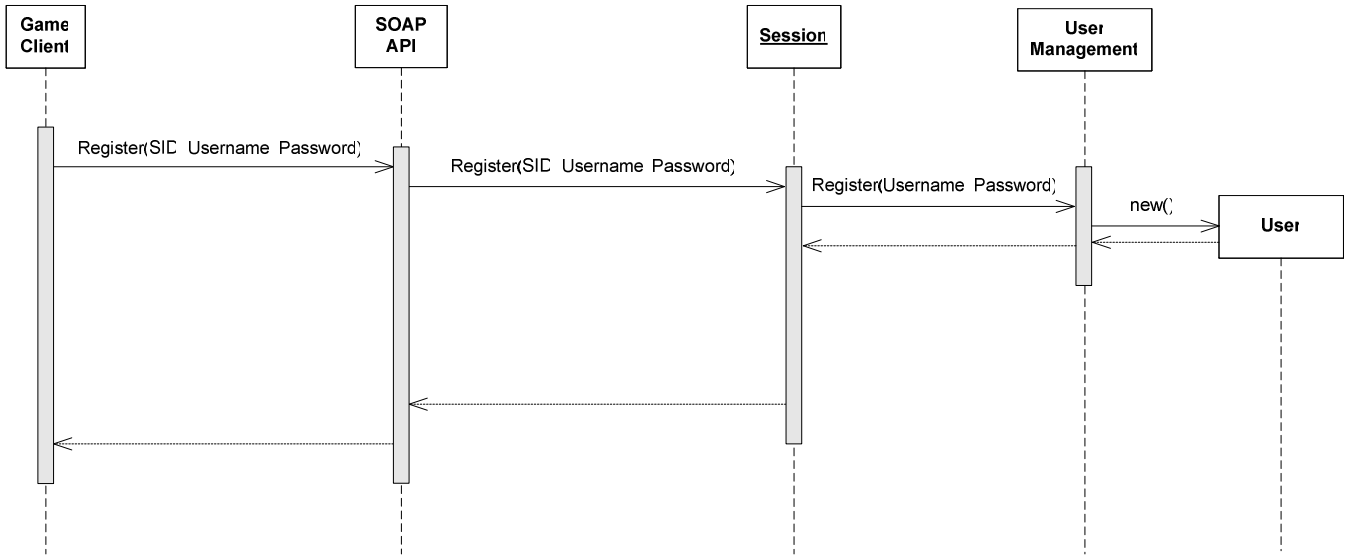

Figure 5: Register operation

6.2.2 Game Persistence

Operations for Game Persistence:

- GetAS (Get Actor Session)
- GetInv (Get Invitation)
- Activate
- Deactivate
- Name

6.2.2.1 GetAS

Name	**GetAS**(SID)
Description	Retrieves detailed information about all assigned ***ActorSessions*** of an ***Actor***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ActorSessions	Vector	Vector of Map **(*1)**

(*1)

Key	Datatype	Value
ASID	String	ActorSession ID
Name	String	ActorSession Name
Actors	String	Comma separated list of Actors
ActorNr	Int	Number in ApplicationInstance
MinAS	Int	Min number of ActorSessions
MaxAS	Int	Max number of ActorSessions
DTCreated	DateTime	Time created (UTC).
State	Int	Application state
Active	Boolean	Indicates whether User is active in ActorSession
NumInQueue	Int	Number of Events in ActorSession EventQueue

Related operations/events	Login, CreateAI, JoinAiRnd, Activate, Deactivate
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.2.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetASRequest">
```

```
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetASResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ActorSessions" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.1.3 Sequence diagram

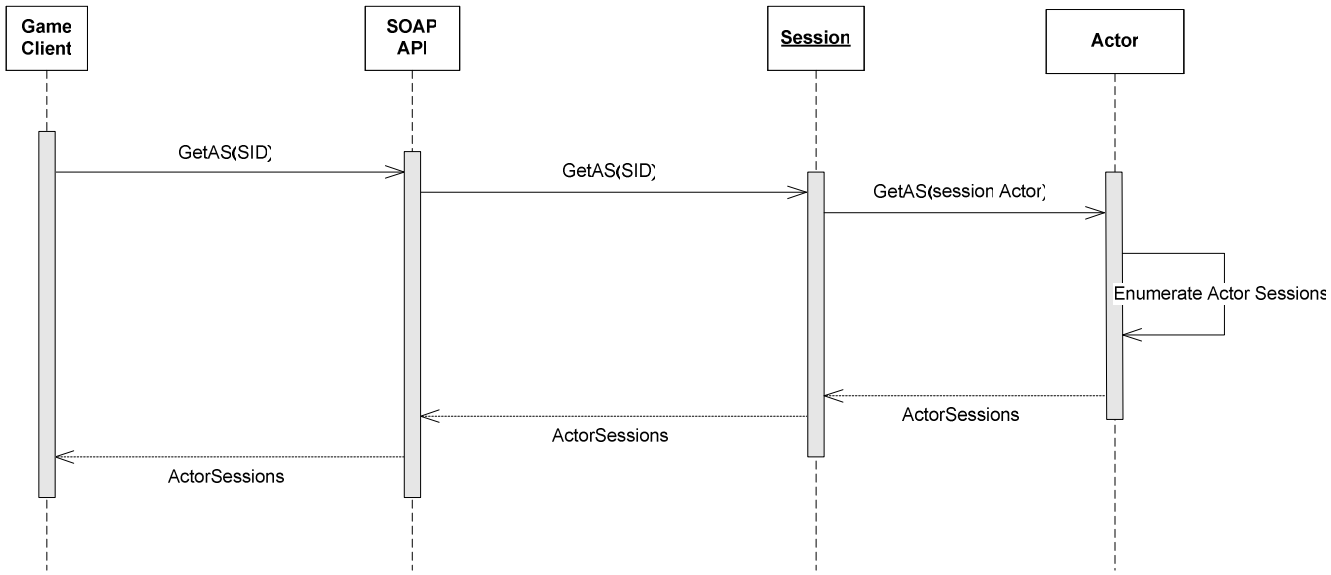

Figure 6: Get Actor Session operation

6.2.2.2 GetInv

Name	**GetInv**(SID)
Description	Get all invitations of an ***Actor***. An Invitation is a special kind of ***ActorSession***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ActorSessions	Vector	Vector of Map **(*1)**

(*1)

Key	Datatype	Value
ASID	String	ActorSession ID
Name	String	ActorSession Name
Actors	String	Comma separated list of Actors
MinAS	Int	Min number of ActorSessions
MaxAS	Int	Max number of ActorSessions
DTCreated	DateTime	Time created (UTC).
ActorNr	Int	Number in ApplicationInstance
NumInGame	Int	Number in Game
NumActive	Int	Number of active ActorSessions in ApplicationInstance
Active	Boolean	Indicates whether ActorSession is active
NumInQueue	Int	Number of Events in ActorSession EventQueue
State	Int	Application state

Related operations/events	CreateAIPrivate
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.2.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetInvRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetInvResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ActorSessions" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.2.3 Sequence diagram

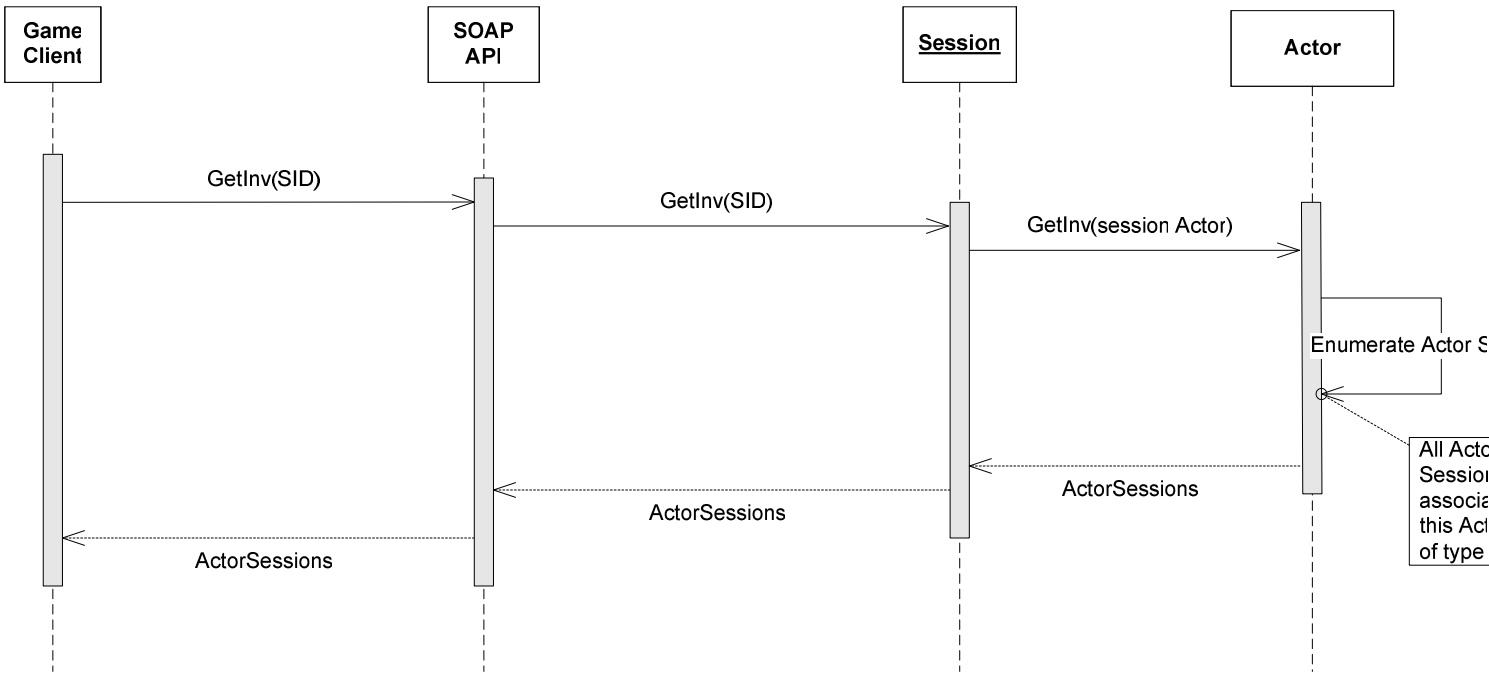

Figure 7: Get invitation operation

6.2.2.3 Activate

Name	**Activate**(SID, ASID)
Description	Activates a specific ***ActorSession***. An EvActivate is sent to all ***ActorSessions*** of the same ***ApplicationInstance***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ASID	String	Valid ActorSession ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ActorNr	Int	Number in ApplicationInstance
Username	String	Actor Name
State	Int	Actor State

Related operations/events	EvActivate
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.2.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="ActivateRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="ActivateResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ActorNr" />
        <xsd:element ref="tns:Username" />
        <xsd:element ref="tns:State" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.3.3 Sequence diagram

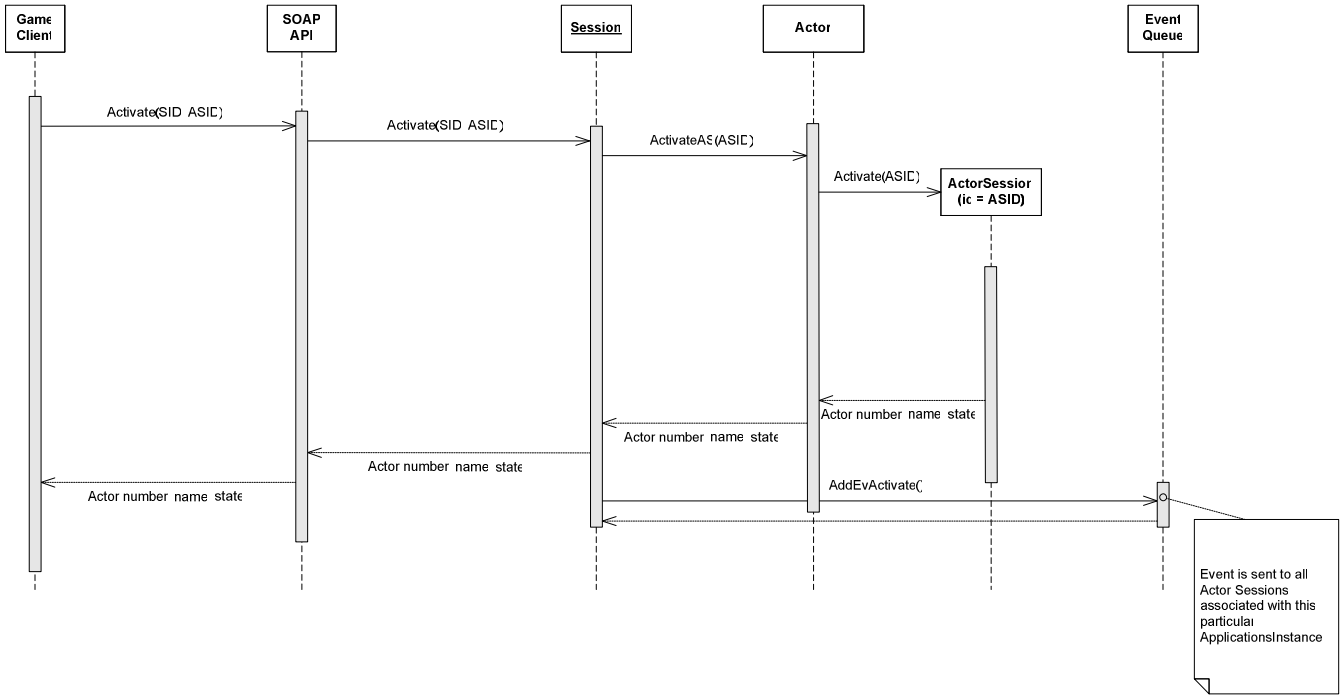

Figure 8: Activate operation

6.2.2.4 Deactivate

Name	**Deactivate**(SID, ASID)
Description	Deactivates a specific ***ActorSession***. An EvDeactivate is sent to all ***ActorSessions*** of the same ***ApplicationInstance***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ASID	String	Valid ActorSession ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id

Related operations/events	EvDeactivate
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.2.4.1 XSD Complex Type: Parameter

```
<xsd:complexType name="DeactivateRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.4.2 XSD Complex Type: Return values

```
<xsd:complexType name="DeactivateResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.4.3 Sequence diagram

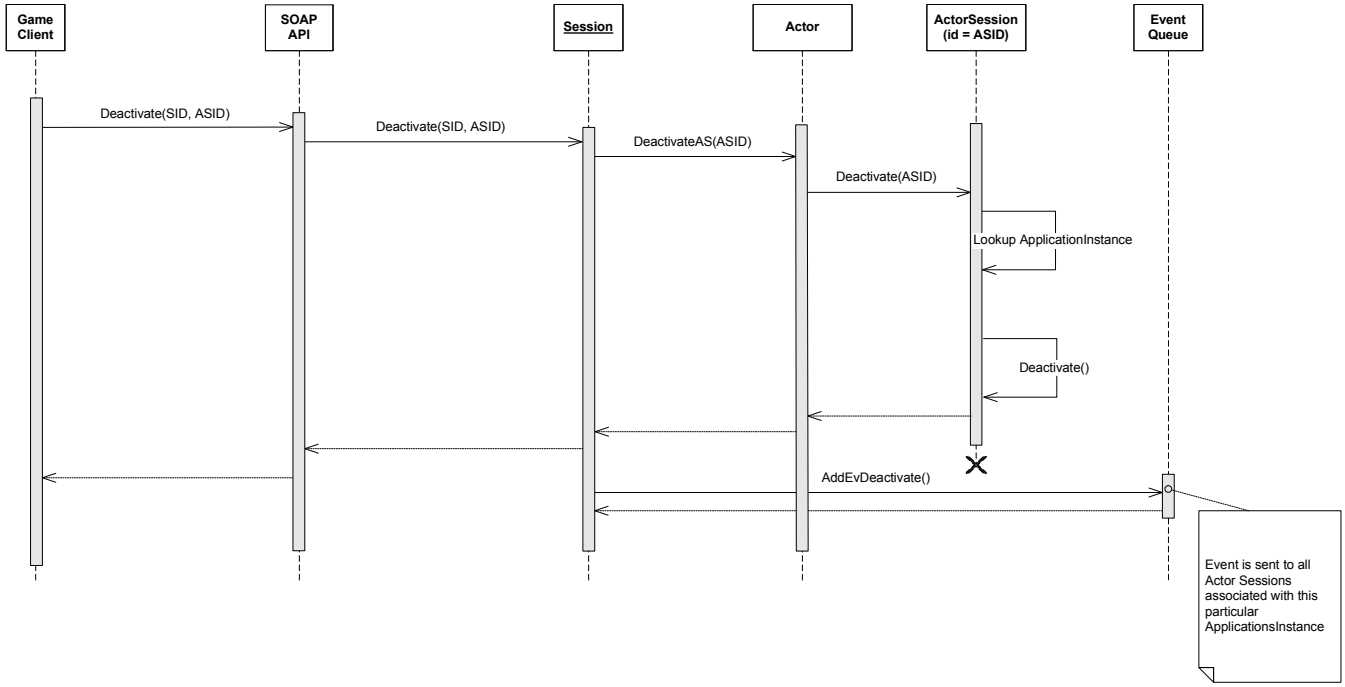

Figure 9: Deactivate operation

6.2.2.5 Name

Name	**Name**(SID, ASID, Name)
Description	Gives a Name to an ***ActorSession***.
Parameter	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**SID**</td><td>String</td><td>Valid Session ID.</td></tr><tr><td>**ASID**</td><td>String</td><td>Valid ActorSession ID.</td></tr><tr><td>**Name**</td><td>String</td><td>Name to use for active ActorSession.</td></tr></table>
Return Values	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**ERR**</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr><tr><td>**DBG**</td><td>String</td><td>Error message (optional)</td></tr><tr><td>**SID**</td><td>String</td><td>Valid session id</td></tr></table>
Related operations/events	CreateAI, CreateAIPrivate, JoinAIRnd
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.2.5.1 XSD Complex Type: Parameter

```
<xsd:complexType name="NameRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ASID" />
        <xsd:element ref="tns:Name" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.5.2 XSD Complex Type: Return values

```
<xsd:complexType name="NameResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.2.5.3 Sequence diagram

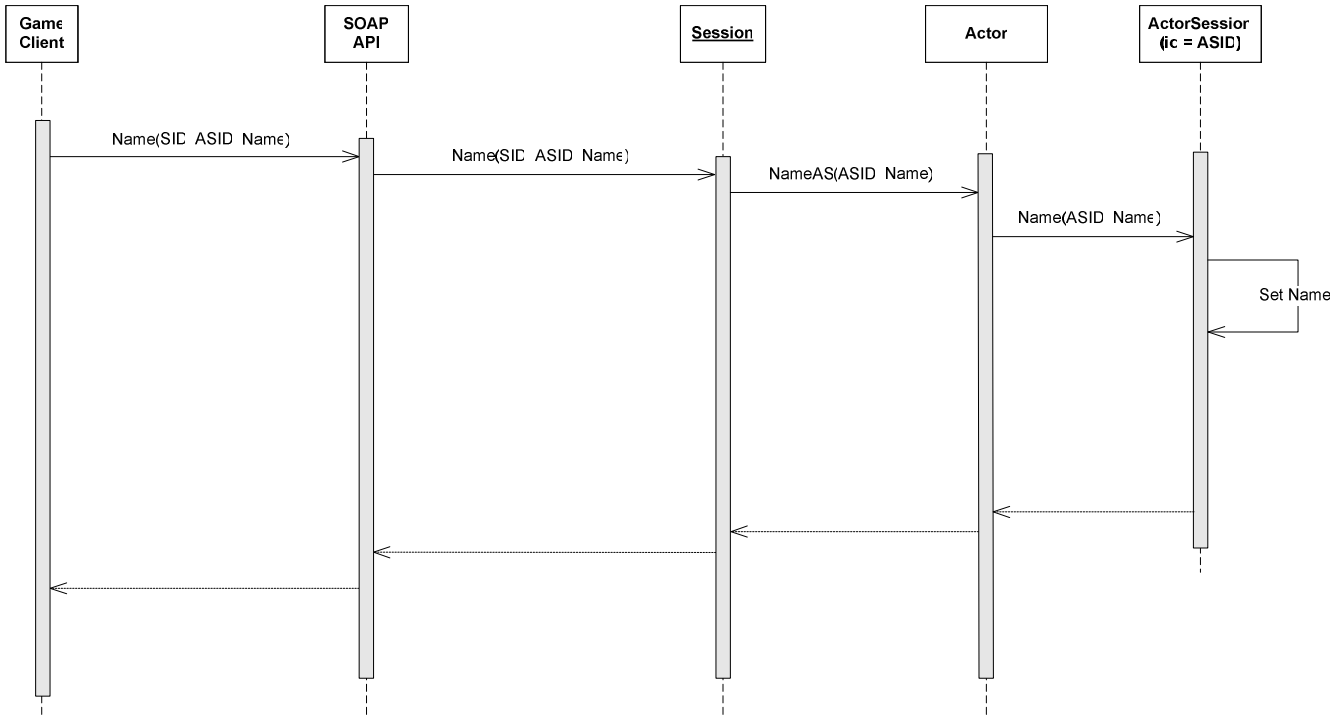

Figure 10: Name Operation

6.2.3 Game Creation and Matchmaking

Operations for:

- CreateAI
- CreateAIPrivate
- ForceAIRnd

6.2.3.1 CreateAI

Name	**CreateAI**(SID, MinAS, MaxAS, Name)
Description	Creates a new public ***ApplicationInstance***. An ***ActorSession*** for the ***Actor*** is automatically created (the ***Actor*** joins the ***ApplicationInstance***).
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID
MinAS	Int	Min number of ActorSessions
MaxAS	Int	Max number of ActorSessions
Name	String	Name of ActorSession

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ASID	String	ActorSession ID.
Actors	Vector	Vector of Map **(*1)**
MinAs	Int	Min number of ActorSessions
MaxAs	Int	Max number of ActorSessions
Name	String	Name of ActorSession

(*1)

Key	Datatype	Value
ActorNr	Int	Number in ApplicationInstance
Username	String	Actor Name
State	Int	Actor State

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.3.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="CreateAIRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="CreateAIResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ASID" />
        <xsd:element ref="tns:Actors" />
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.1.3 Sequence diagram

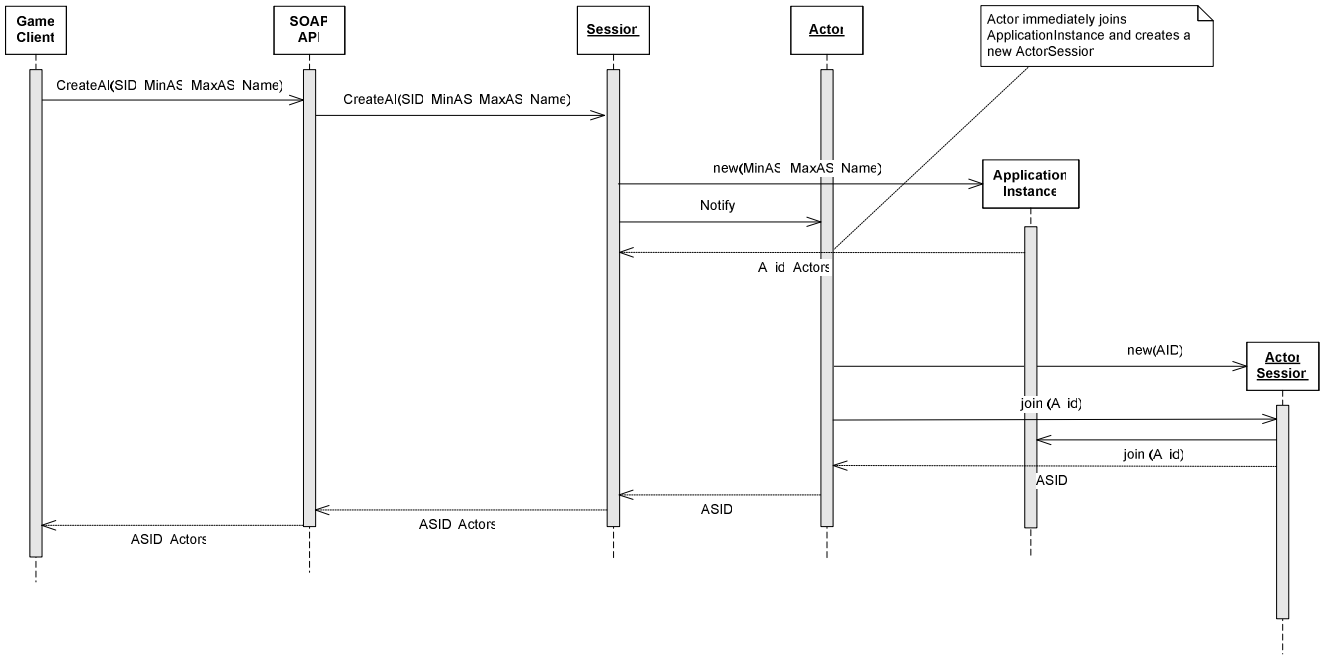

Figure 11: Create Application Instance Operation

6.2.3.2 CreateAIPrivate

Name	**CreateAIPrivate**(SID, Actors, MinAS, MaxAS, Name)
Description	Creates a private ***ApplicationInstance***. An ***ActorSession*** for the calling ***Actor*** is created. ***ActorSessions*** for all invited ***Actors*** are created. These are marked as Invitations.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID
Actors	Vector	Vector of String
MinAS	Int	Min number of ActorSessions
MaxAS	Int	Max number of ActorSessions
Name	String	ActorSession Name

<table>
<tr><td>Return Values</td><td>
<table>
<tr><th>Key</th><th>Datatype</th><th>Value</th></tr>
<tr><td>ERR</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr>
<tr><td>DBG</td><td>String</td><td>Error message (optional)</td></tr>
<tr><td>SID</td><td>String</td><td>Valid session id</td></tr>
<tr><td>ASID</td><td>String</td><td>ActorSession ID.</td></tr>
<tr><td>Actors</td><td>Vector</td><td>Vector of Map (*1)</td></tr>
<tr><td>MinAS</td><td>Int</td><td>Min number of ActorSessions</td></tr>
<tr><td>MaxAS</td><td>Int</td><td>Max number of ActorSessions</td></tr>
<tr><td>Name</td><td>String</td><td>ActorSession Name</td></tr>
<tr><td>ActorNr</td><td>Int</td><td>Actor’s own actor number</td></tr>
</table>
(*1)
<table>
<tr><th>Key</th><th>Datatype</th><th>Value</th></tr>
<tr><td>ActorNr</td><td>Int</td><td>Number in ApplicationInstance</td></tr>
<tr><td>Username</td><td>String</td><td>Actor Name</td></tr>
<tr><td>State</td><td>Int</td><td>Actor State</td></tr>
</table>
</td></tr>
<tr><td>Related operations/events</td><td>Activate, GetInv</td></tr>
<tr><td>Errors</td><td>SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTOR_NOT_FOUND, PARAMETER_OUT_OF_RANGE</td></tr>
</table>

6.2.3.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="CreateAIPrivateRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Actors" />
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="CreateAIPrivateResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ASID" />
        <xsd:element ref="tns:Actors" />
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
        <xsd:element ref="tns:ActorNr" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.2.3 Sequence diagram

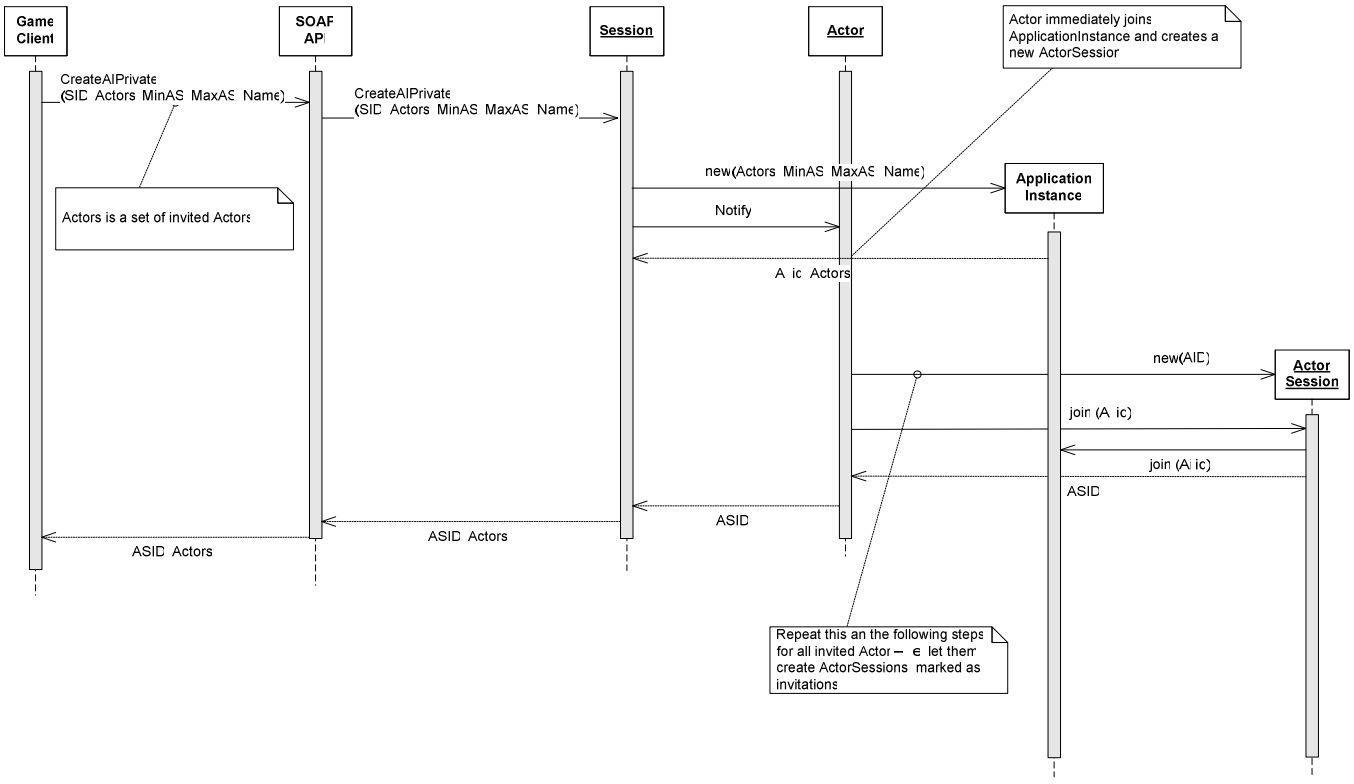

Figure 12: Create a Private Application instance operation

6.2.3.3 ForceAIRnd

Name	ForceAIRnd (SID, MinAS, MaxAS, Name)
Description	The ***Actor*** joins randomly an open public ***ApplicationInstance***. The ***ApplicationInstance*** to join is decided by the platform. An EvJoinAIRnd is sent to all Actors attached to the AI.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID
MinAS	Int	Min number of ActorSessions
MaxAS	Int	Max number of ActorSessions
Name	String	Name of ActorSession

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ASID	String	ActorSession ID.
Actors	Vector	Vector of Map **(*1)**
MinAs	Int	Min number of ActorSessions
MaxAs	Int	Max number of ActorSessions
Name	String	Name of ActorSession
ActorNr	Int	Actor's own actor number

(*1)

Key	Datatype	Value
ActorNr	Int	Number in ApplicationInstance
Username	String	Actor Name
State	Int	Actor State

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, PARAMETER_OUT_OF_RANGE, OPEN_APPLICATIONINSTANCE_NOT_FOUND

6.2.3.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="ForceAIRndRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="ForceAIRndResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ASID" />
        <xsd:element ref="tns:Actors" />
        <xsd:element ref="tns:MinAS" />
        <xsd:element ref="tns:MaxAS" />
        <xsd:element ref="tns:Name" />
        <xsd:element ref="tns:ActorNr" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.3.3.3 Sequence diagram

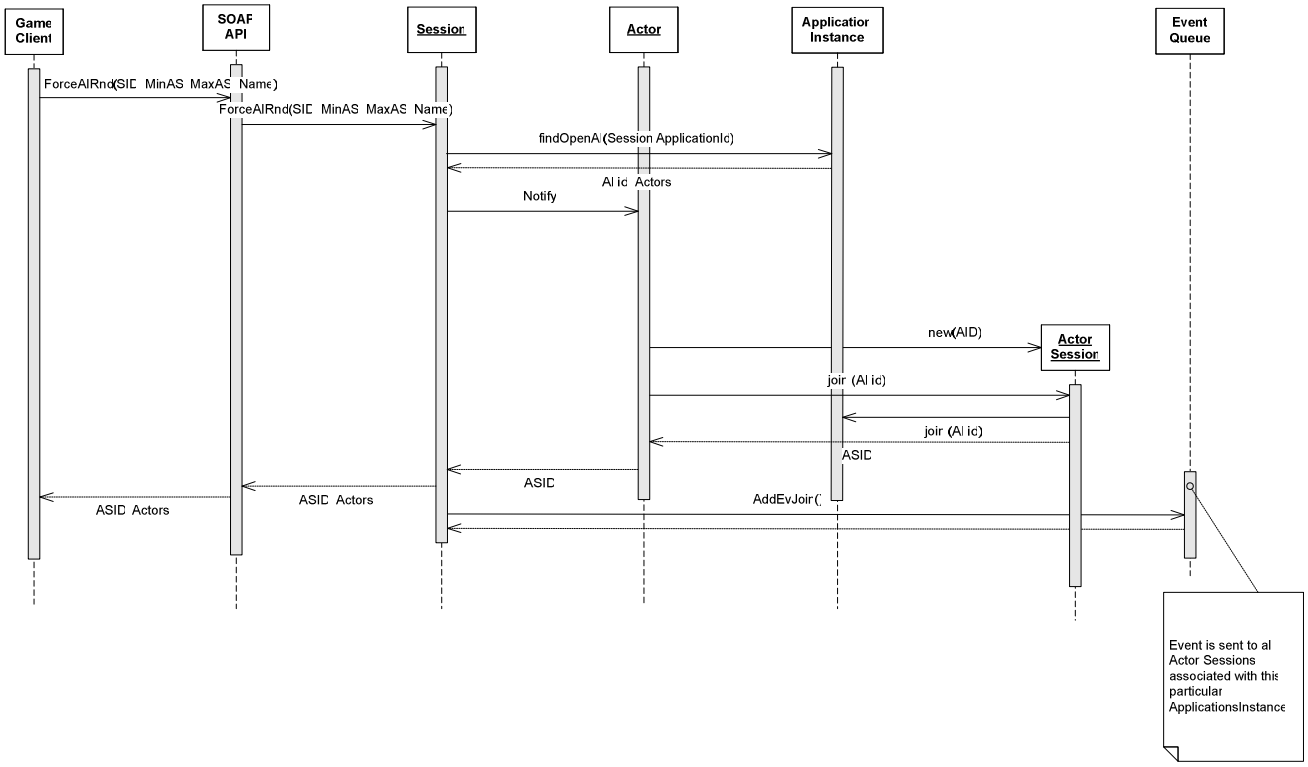

Figure 13: Actor joins a randomly selected Application instance operation

6.2.4 Gameplay

Operations for Gameplay:

- Start
- End
- Quit
- HandoverTurn
- SendData

6.2.4.1 Start

Name	**Start(**SID, ASID)		
Description	Starts the ***ApplicationInstance***. An EvStart is sent to all connected Actors.		
Parameter	Key	Datatype	Value
	SID	String	Valid Session ID
	ASID	String	Valid ActorSession ID
Return Values	/		
Related operations/events	EvStart		
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY		

6.2.4.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="StartRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="StartResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.1.3 Sequence diagram

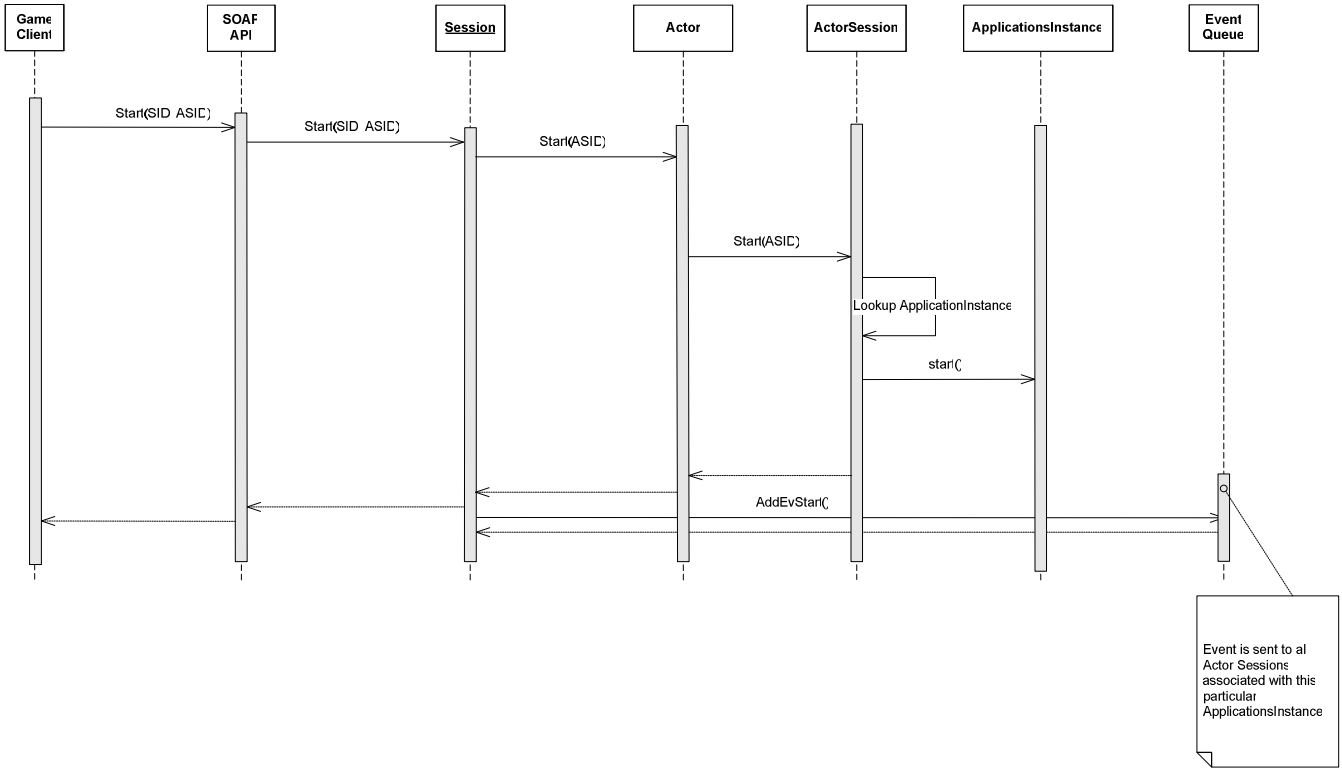

Figure 14: Start operation

6.2.4.2 End

Name	**End**(SID, ASID)
Description	Ends the ***ApplicationInstance***. An EvEnd is sent to all connected Actors.
Parameter	Key / Datatype / Value **SID** / String / Valid Session ID. **ASID** / String / Valid ActorSession ID.
Return Values	/
Related operations/events	EvEnd
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.4.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="EndRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="EndResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.2.3 Sequence diagram

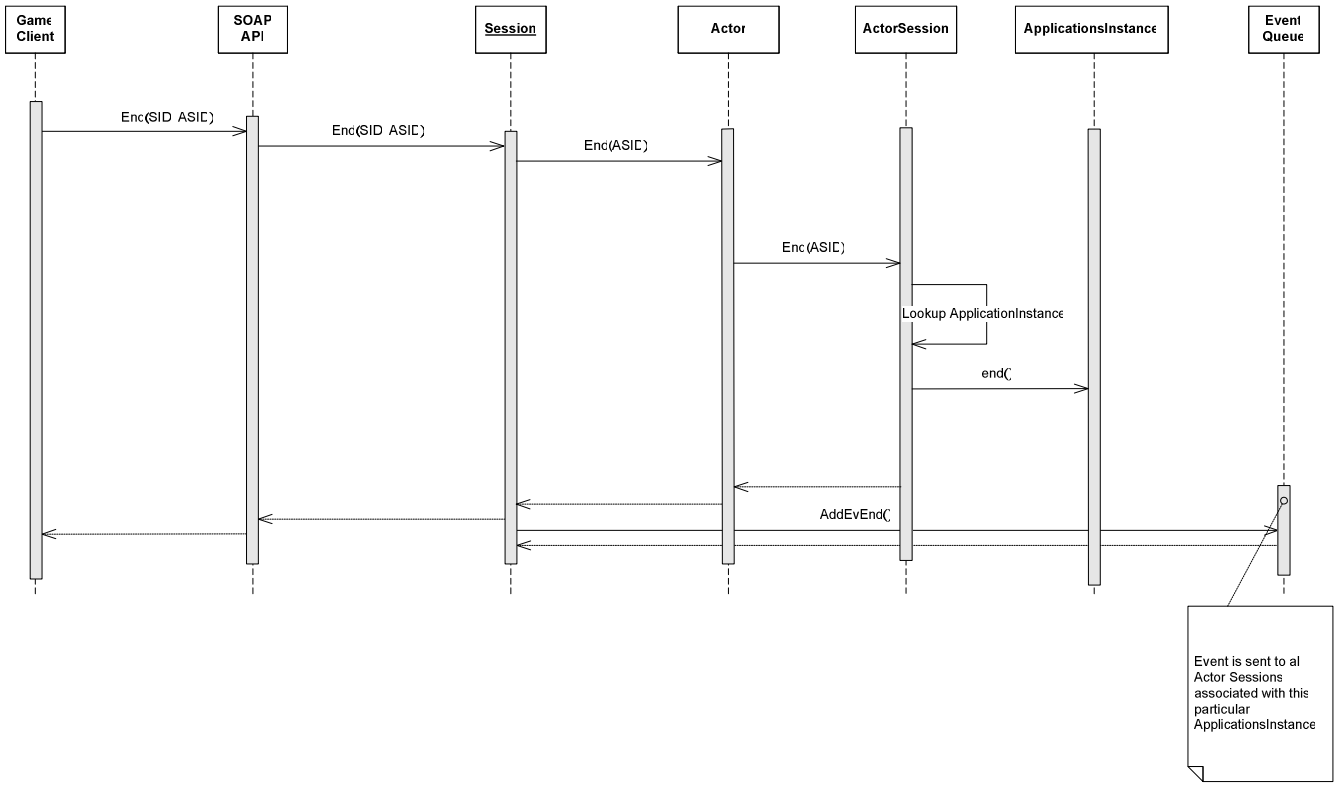

Figure 15: End operation

6.2.4.3 Quit

Name	**Quit**(SID, ASID)
Description	Quits the ***ActorSessions***. Important: The ***ApplicationInstance*** is NOT ended. An EvQuit is sent to all ActorSessions.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ASID	String	Valid ActorSession ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id

Related operations/events	EvQuit
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.4.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="QuitRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="QuitResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.3.3 Sequence diagram

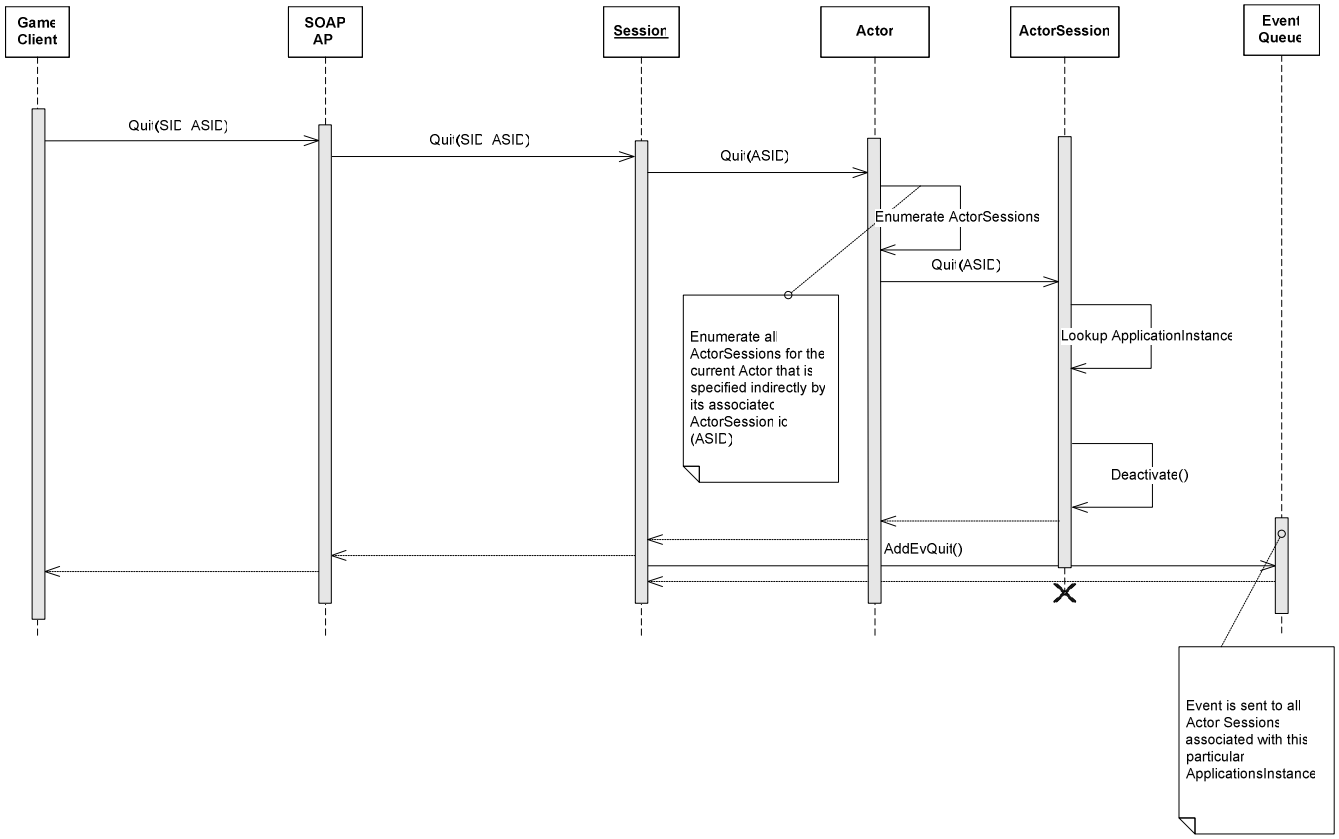

Figure 16: Quit Operation

6.2.4.4 HandoverTurn

Name	**HandoverTurn**(SID, ASID, Data, ActorNr)
Description	Signals the handover of the turn to all connected ***Actors***. An EvTurn is sent to all connected ***Actors***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID
ASID	String	Valid ActorSession ID
Data	AnyType	Turn Data
ActorNr	Int	Target ActorNr

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id

Related operations/events	EvTurn
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.4.4.1 XSD Complex Type: Parameter

```
<xsd:complexType name="HandoverTurnRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Data" />
        <xsd:element ref="tns:ActorNr" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.4.2 XSD Complex Type: Return values

```
<xsd:complexType name="HandoverTurnResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.4.3 Sequence diagram

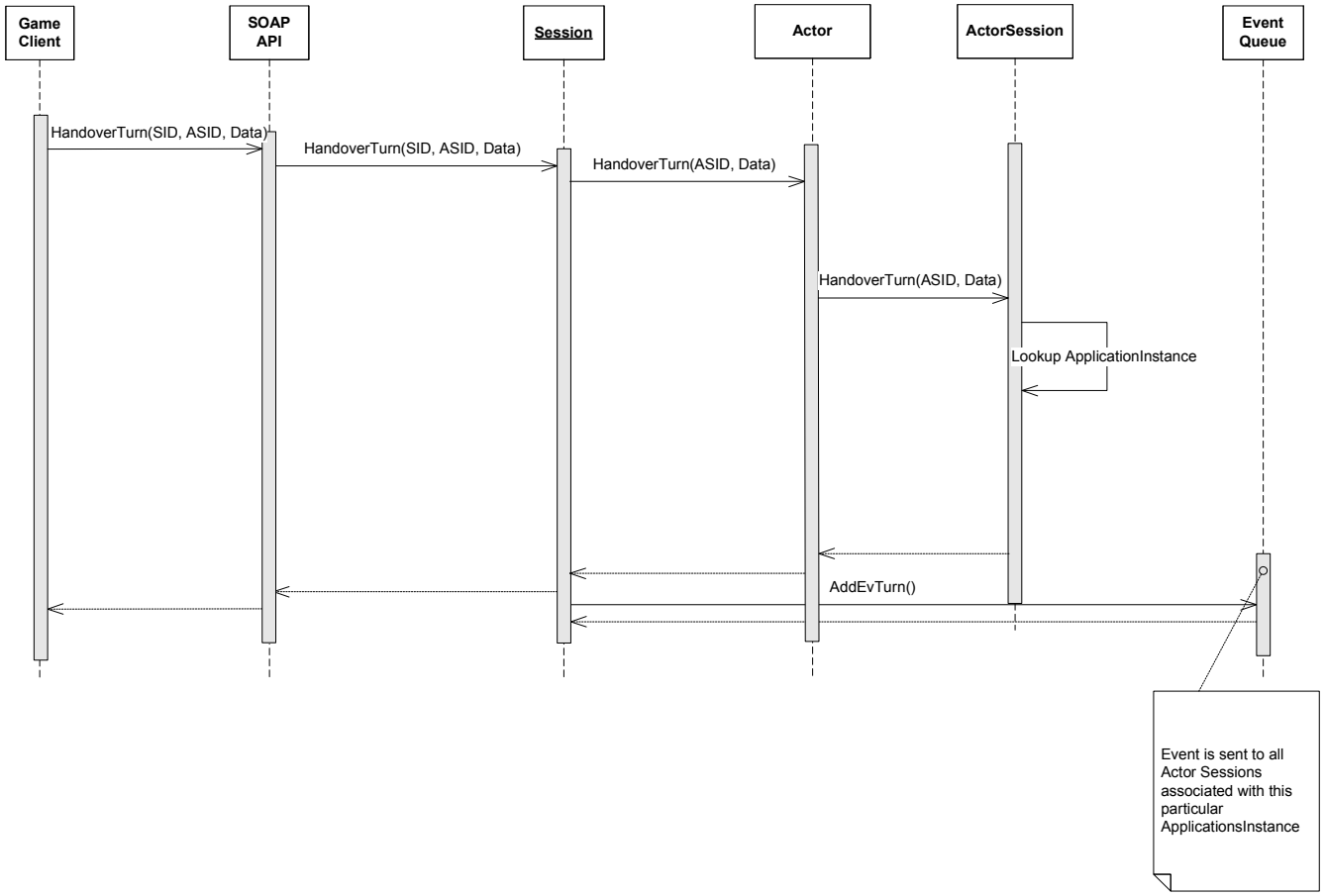

Figure 17: Handover Turn operation

6.2.4.5 SendData

Name	**SendData**(SID, ASID, Data, Actors)
Description	Sends data to all connected Actors. An EvData is sent to all connected ***Actors***.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID
ASID	String	Valid ActorSession ID
Data	AnyType	Data
Actors	Vector	Vector of String

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id

Related operations/events	EvData
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.4.5.1 XSD Complex Type: Parameter

```
<xsd:complexType name="SendDataRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Data" />
        <xsd:element ref="tns:Actors" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.5.2 XSD Complex Type: Return values

```
<xsd:complexType name="SendDataResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.4.5.3 Sequence diagram

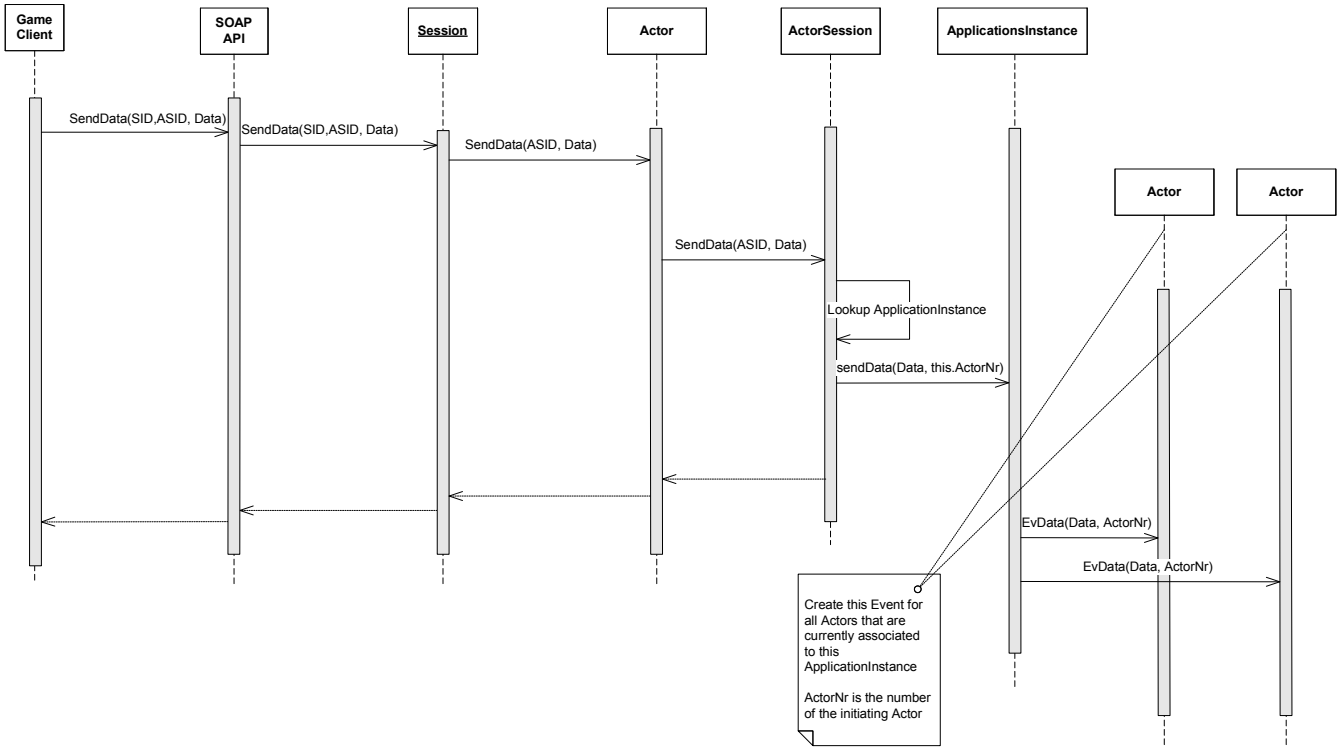

Figure 18: Send Data operation

6.2.5 Messaging

Operations for Messaging:

- SendTxtMsg

6.2.5.1 SendTxtMsg

Name	**SendTxtMsg**(SID, ASID, Msg, Actors)		
Description	Sends a text message to all connected ***Actors***.		
Parameter	Key	Datatype	Value
	SID	String	Valid Session ID
	ASID	String	Valid ActorSession ID
	Msg	String	Text Message
	Actors	Vector	Vector of String
Return Values	Key	Datatype	Value
	ERR	Int	Errorcode (0 if no error occurred)
	DBG	String	Error message (optional)
	SID	String	Valid session id
Related operations/events	EvTxtMsg		
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY		

6.2.5.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="SendTxtMsgRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Msg" />
        <xsd:element ref="tns:Actors" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.5.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="SendTxtMsgResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.5.1.3 Sequence diagram

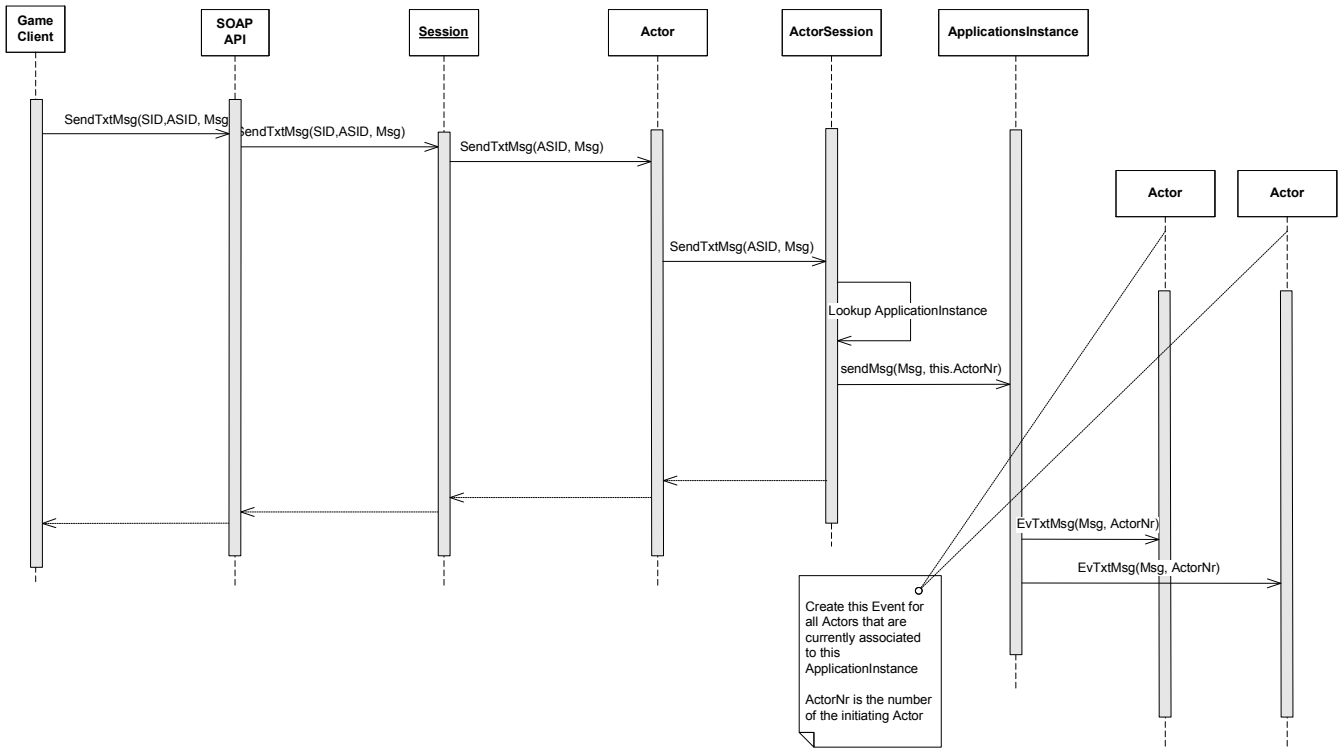

Figure 19: Send a Text message to all Actors operation

6.2.6 Highscore

Operations for:

- SetScore
- GetScorePos
- GetScoresByPos
- GetScoresByID
- GetScoresByUser

6.2.6.1 SetScore

Name	**SetScore**(SID, ScoreTable, Score)
Description	Send a Score to the server.
Parameter	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**SID**</td><td>String</td><td>Valid Session ID</td></tr><tr><td>**ScoreTable**</td><td>String</td><td>Score Table ID</td></tr><tr><td>**Score**</td><td>Long</td><td>Highscore to set</td></tr></table>
Return Values	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**ERR**</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr><tr><td>**DBG**</td><td>String</td><td>Error message (optional)</td></tr><tr><td>**SID**</td><td>String</td><td>Valid session id</td></tr><tr><td>**ScoreID**</td><td>String</td><td>Unique score id.</td></tr></table>
Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, SCORE_TABLE_NOT_FOUND

6.2.6.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="SetScoreRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreTable" />
        <xsd:element ref="tns:Score" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="SetScoreResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreID" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.1.3 Sequence diagram

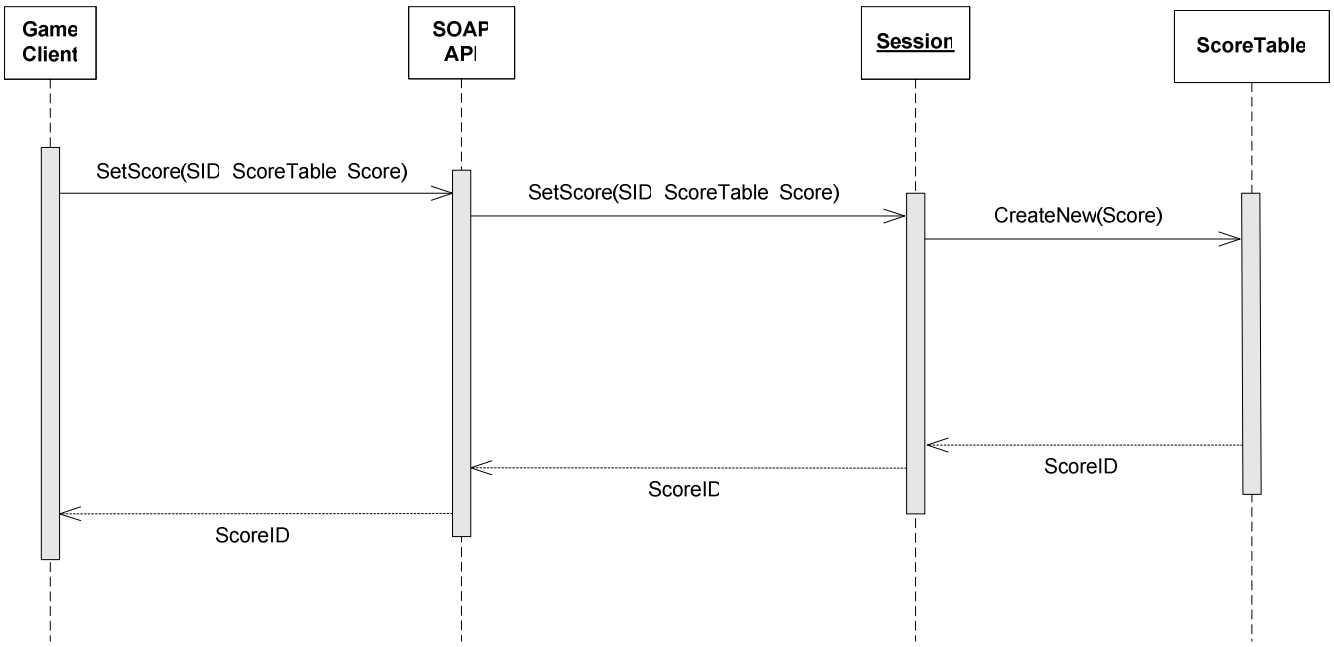

Figure 20: Set Score operation

6.2.6.2 GetScorePos

Name	**GetScorePos**(SID, ScoreID, TimeFrame)
Description	Get the position of a Score with the ScoreID.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ScoreID	String	Unique score ID.
TimeFrame	String	d = day today w = week m = month y = year o = overall

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
Pos	Int	Position in Highscore list.

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED

6.2.6.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetScorePosRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreID" />
        <xsd:element ref="tns:TimeFrame" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetScorePosResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Pos" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.2.3 Sequence diagram

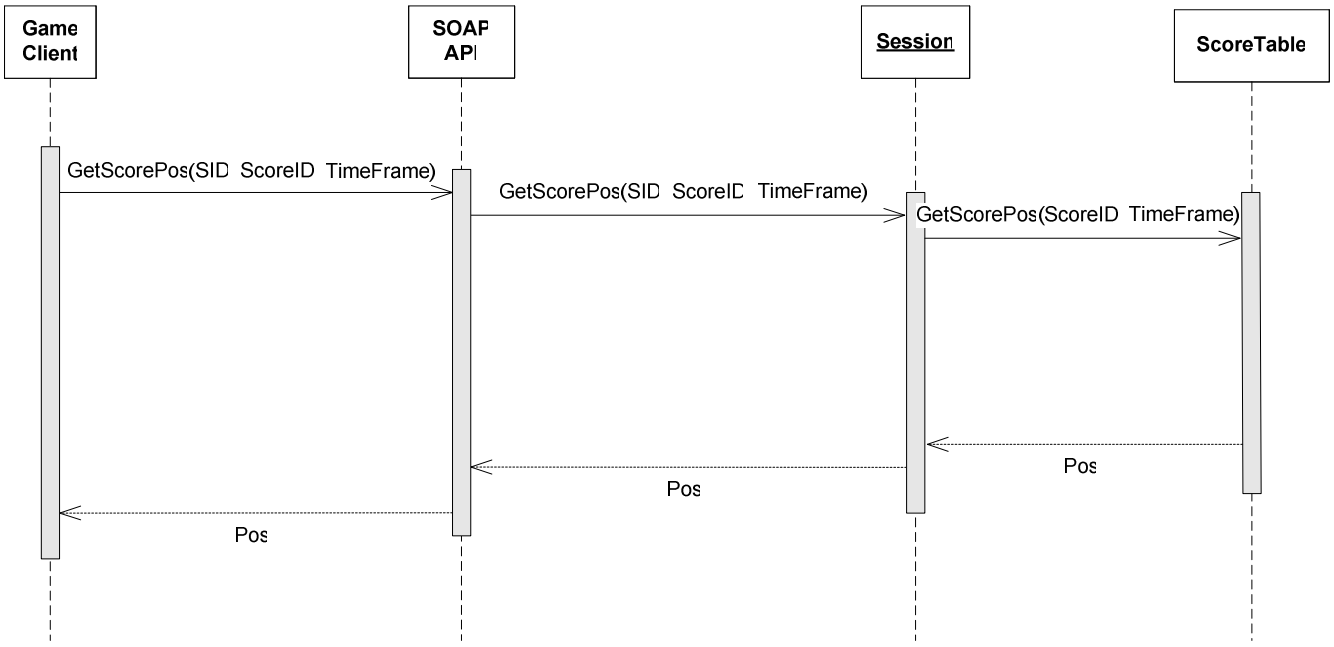

Figure 21: Get a position for the score Operation

6.2.6.3 GetScoresByPos

Name	**GetScoresByPos**(SID, ScoreTable, Pos, Count, Filter, TimeFrame)
Description	Retrieves a certain number of scores from a specific position.

Parameter

Key	Datatype	Value
SID	String	Valid Session ID.
ScoreTable	String	Unique score table ID.
Pos	Int	Position to start.
Count	Int	Number of scores.
Filter	Int	0: all scores 1: my scores
TimeFrame	String	d = day today w = week m = month y = year o = overall

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
Scores	Vector	Vector of Map **(*1)**

(*1)

Key	Datatype	Value
ScoreID	String	Unique score ID.
Pos	Int	Position in highscore list.
Username	String	Username.
Country	String	Country code.
Score	Long	Score.
DTCreated	DateTime	Timestamp created (UTC).

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, SCORE_TABLE_NOT_FOUND, HIGHSCORE_MAXPOS_EXCEEDED, HIGHSCORE_NOT_FOUND

6.2.6.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetScoresByPosRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreTable" />
        <xsd:element ref="tns:Pos" />
        <xsd:element ref="tns:Count" />
        <xsd:element ref="tns:Filter" />
        <xsd:element ref="tns:TimeFrame" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetScoresByPosResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Scores" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.3.3 Sequence diagram

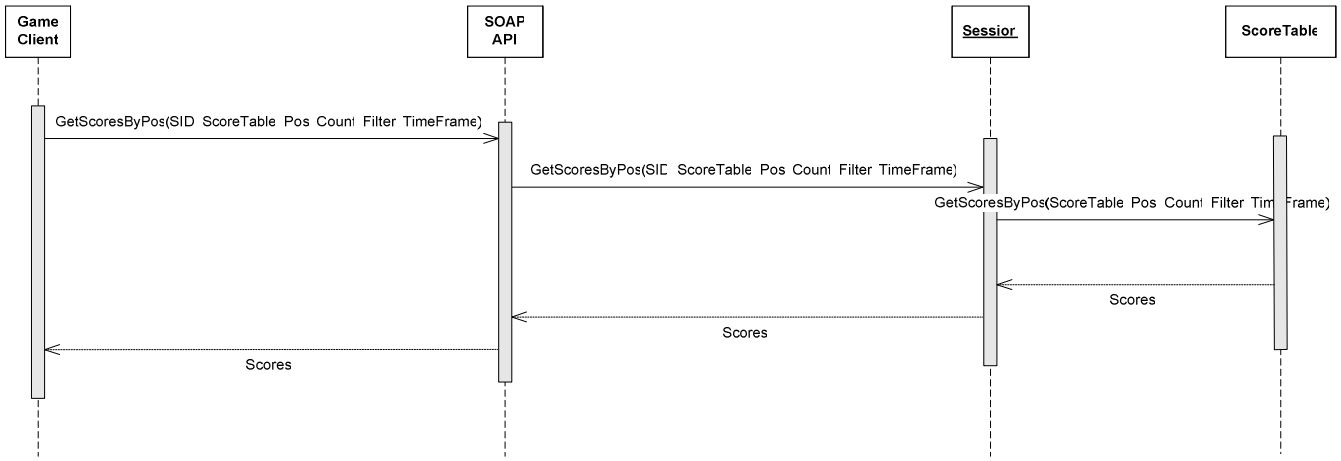

Figure 22: Retrieve a certain number of scores from a position operation

6.2.6.4 GetScoresByID

Name	**GetScoresByID**(SID, ScoreID, Count, Offset, Filter, TimeFrame)
Description	Retrieves a certain number of scores for a specific ID.

Parameter

Key	Datatype	Value
SID	String	Valid Session ID.
ScoreID	String	Unique score ID.
Count	Int	Number of scores to retrieve.
Offset	Int	The Offset to start from.
Filter	Int	0: all scores 1: my scores
TimeFrame	String	d = day today w = week m = month y = year o = overall

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
Scores	Vector	Vector of Map **(*1)**

(*1)

Key	Datatype	Value
ScoreID	String	Unique score ID.
Pos	Int	Position in highscore list.
Username	String	Username.
Country	String	Country code.
Score	Long	Score.
DTCreated	DateTime	Timestamp created (UTC).

Related operations/events	/

Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, HIGHSCORE_MAXPOS_EXCEEDED, HIGHSCORE_NOT_FOUND

6.2.6.4.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetScoresByIDRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreID" />
        <xsd:element ref="tns:Count" />
        <xsd:element ref="tns:Offset" />
        <xsd:element ref="tns:Filter" />
        <xsd:element ref="tns:TimeFrame" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.4.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetScoresByIDResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Scores" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.4.3 Sequence diagram

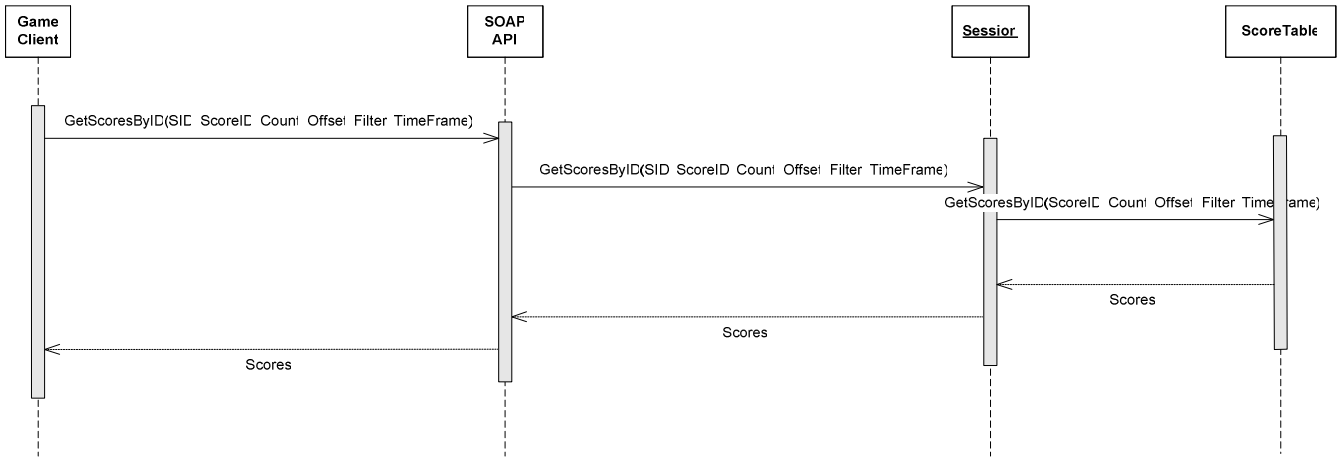

Figure 23: Retrieve a certain number of scores for a certain ID Operation

6.2.6.5 GetScoresByUser

Name	**GetScoresByUser**(SID, ScoreTable, Username, Count, Offset, Filter, TimeFrame)
Description	Retrieves a certain number of scores for a specific username.

Parameter

Key	Datatype	Value
SID	String	Valid Session ID.
ScoreTable	String	Unique score table ID.
Username	String	Username (unique).
Count	Int	Number of scores to retrieve.
Offset	Int	The Offset to start from.
Filter	Int	0: all scores 1: my scores
TimeFrame	String	d = day today w = week m = month y = year o = overall

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
Scores	Vector	Vector of Map **(*1)**

(*1)

Key	Datatype	Value
ScoreID	String	Unique score ID.
Pos	Int	Position in highscore list.
Username	String	Username.
Country	String	Country code.
Score	Long	Score.
DTCreated	DateTime	Timestamp created (UTC).

Related operations/events	/

Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, SCORE_TABLE_NOT_FOUND, HIGHSCORE_MAXPOS_EXCEEDED, HIGHSCORE_NOT_FOUND

6.2.6.5.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetScoresByUserRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreTable" />
        <xsd:element ref="tns:Username" />
        <xsd:element ref="tns:Count" />
        <xsd:element ref="tns:Offset" />
        <xsd:element ref="tns:Filter" />
        <xsd:element ref="tns:TimeFrame" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.5.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetScoresByUserResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Scores" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.6.5.3 Sequence diagram

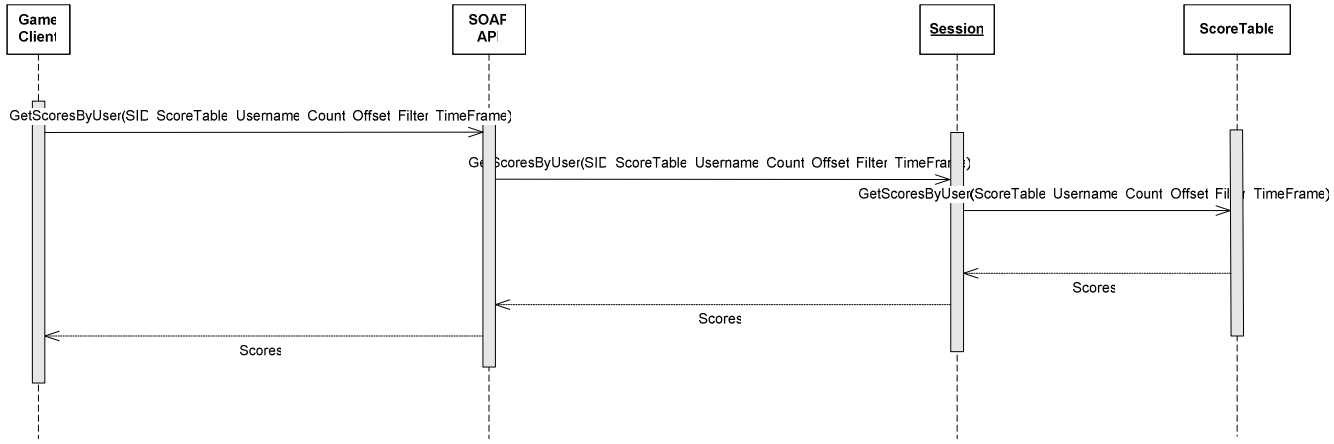

Figure 24: Get a certain number of scores for a specific user name Operation

6.2.7 Shadows (Ghosts)

Operations for:

- SetShadow
- GetShadow

6.2.7.1 SetShadow

Name	**SetShadow**(SID, ScoreTable, Score, Shadow, TotalBytes)
Description	Send a Shadow and Score to the platform.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ScoreTable	String	Unique score table ID.
Score	Long	Highscore to set.
Shadow	Binary	Shadow data. It's associated with the Score.
TotalBytes	Int	Length of Shadow binary data

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
ScoreID	String	Unique score id.

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, SCORE_TABLE_NOT_FOUND

6.2.7.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="SetShadowRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreTable" />
        <xsd:element ref="tns:Score" />
        <xsd:element ref="tns:Shadow" />
        <xsd:element ref="tns:TotalBytes" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.7.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="SetShadowResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreID" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.7.1.3 Sequence diagram

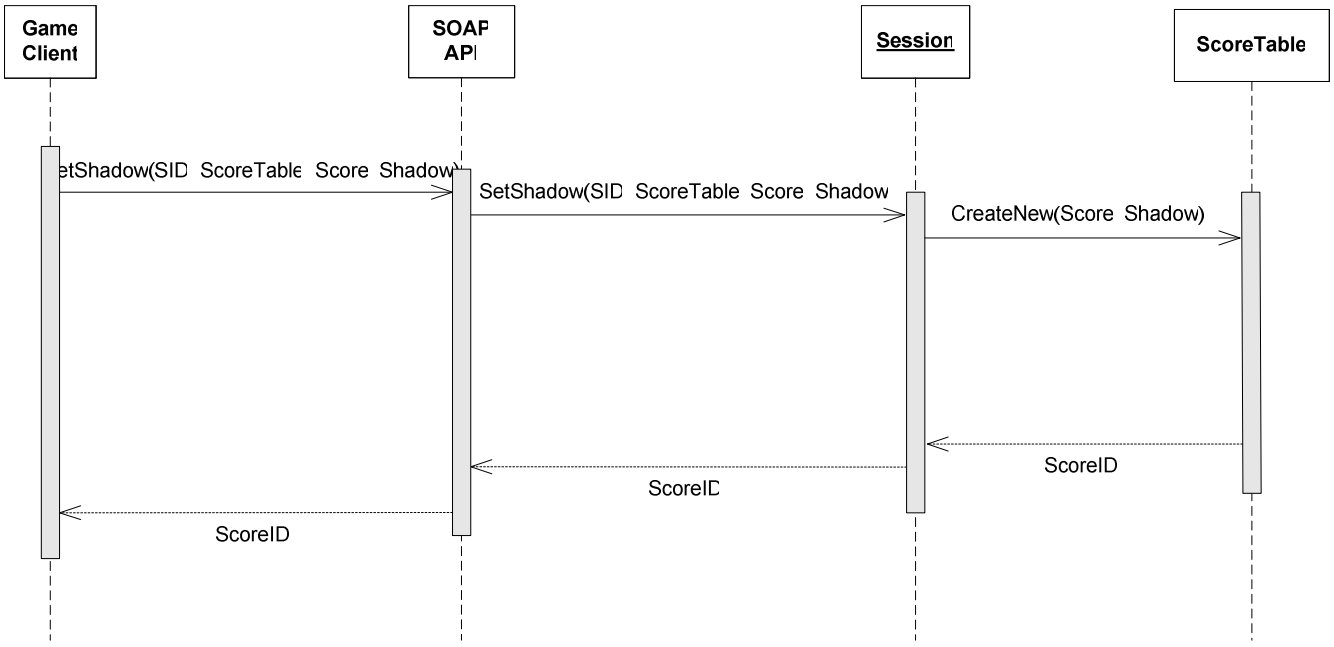

Figure 25: Set Shadow Operation

6.2.7.2 GetShadow

Name	**GetShadow**(SID, ScoreID)
Description	Retrieve a Score from the platform.
Parameter	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**SID**</td><td>String</td><td>Valid Session ID.</td></tr><tr><td>**ScoreID**</td><td>String</td><td>Unique ScoreID.</td></tr></table>
Return Values	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**ERR**</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr><tr><td>**DBG**</td><td>String</td><td>Error message (optional)</td></tr><tr><td>**SID**</td><td>String</td><td>Valid session id</td></tr><tr><td>**Shadow**</td><td>Binary</td><td>Shadow data.</td></tr></table>
Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, HIGHSCORE_NOT_FOUND

6.2.7.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetShadowRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:ScoreID" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.7.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetShadowResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Shadow" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.7.2.3 Sequence diagram

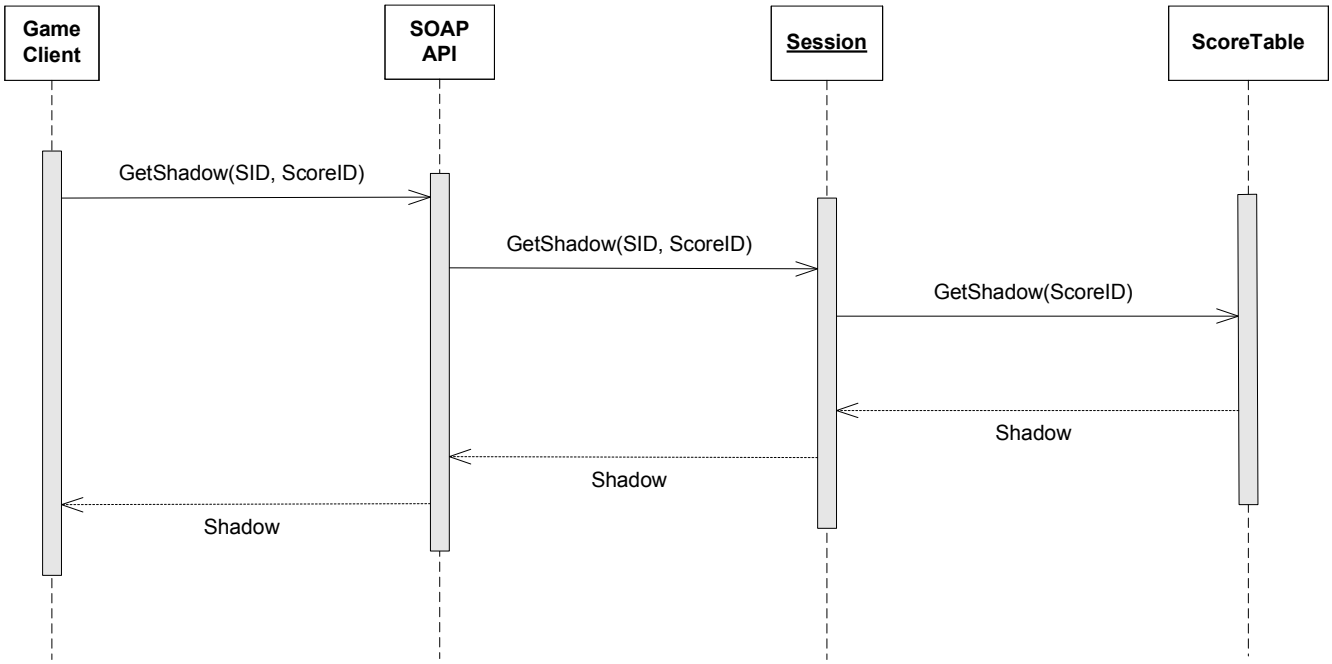

Figure 26: Get Shadow operation

6.2.8 Misc

Operation for:

- Ping
- DateTime

6.2.8.1 Ping

Name	**Ping**(SID)		
Description	Instantly returns to measure the Ping time.		
Parameter	Key	Datatype	Value
	SID	String	Valid Session ID.
Return Values	Key	Datatype	Value
	ERR	Int	Errorcode (0 if no error occurred)
	DBG	String	Error message (optional)
	SID	String	Valid session id
Related operations/events	/		
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED		

6.2.8.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="PingRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.8.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="PingResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.8.1.3 Sequence diagram

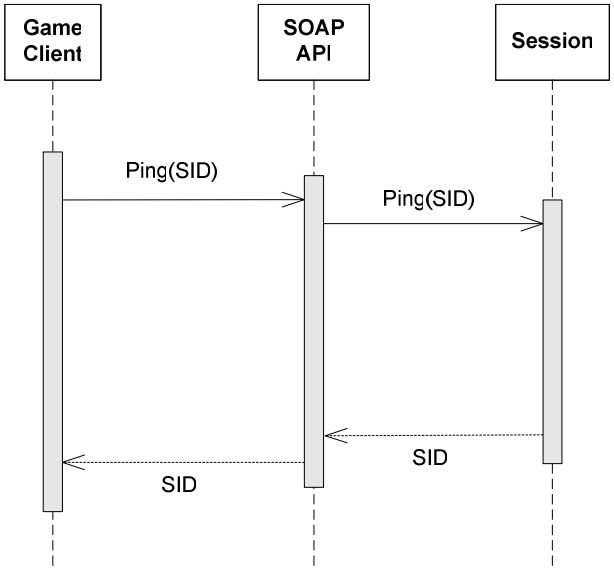

Figure 27: Ping Operation

6.2.8.2 DateTime

Name	**DateTime**(SID)		
Description	Get the current server time and date (UTC).		
Parameter	Key	Datatype	Value
	SID	String	Valid Session ID.
Return Values	Key	Datatype	Value
	ERR	Int	Errorcode (0 if no error occurred)
	DBG	String	Error message (optional)
	SID	String	Valid session id
	DateTime	DateTime	Current UTC time.
Related operations/events	/		
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED		

6.2.8.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="DateTimeRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.8.2.2 XSD Complex Type: Return values

```
<xsd:complexType name="DateTimeResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:SessionCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:DateTime" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.8.2.3 Sequence diagram

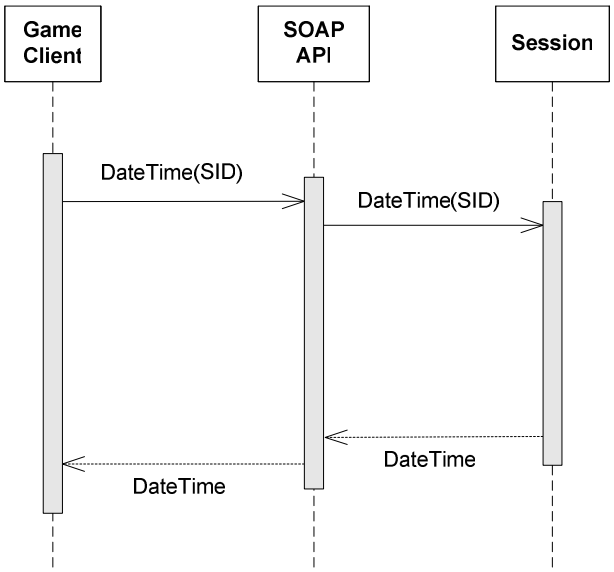

Figure 28: Get DateTime Operation

6.2.9 Event Queue Operations

The polling of events is necessary because in the MIDP 1.0 [MIDP1.0] and MIDP 2.0 [MIDP2.0] specification, only HTTP is mandatory supported.

Communication Protocol	Type	MIDP 1.0	MIDP 2.0
HTTP	Client/Server	Mandatory	Mandatory
HTTPS	Client/Server	-	Mandatory
TLS (SSL)	Client/Server	-	Optional
TCP	Client/Server	-	Optional
UDP	Client/Server	-	Optional
Serial Cable	P2P (2 players)	-	Optional
Infrared	P2P (2 players)	-	Optional
Bluetooth	P2P (2-8 players)	-	Optional
SMS	P2P or Client/Server	Optional: JSR-120	Optional: JSR-120
MMS	P2P or Client/Server	Optional: JSR-205	Optional: JSR-205

Source: Forum Nokia (http://forum.nokia.com)

Figure 29: MIDP communication protocols

Operations for Event Queue Operations:

- GetEvents
- GetEventHistory
- RaiseEvent

6.2.9.1 GetEvents

Name	**GetEvents**(SID, ASID)
Description	Retrieves events from the event queue in the ***ActorSession*** *specified by the ASID*.
Parameter	

Key	Datatype	Value
SID	String	Valid Session ID.
ASID	String	ActorSession ID.

Return Values

Key	Datatype	Value
ERR	Int	Errorcode (0 if no error occurred)
DBG	String	Error message (optional)
SID	String	Valid session id
Events	Vector	Vector of Map **(*1)**

(*1) Key	Datatype	Value
Code	String	Event Message
ActorNr	Int	Actor number who initiated the event
Data	Anytype	Event parameters according to code value

Related operations/events	/
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.9.1.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetEventsRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandRequest">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.1.2 XSD Complex Type: Return values

```
<xsd:complexType name="GetEventsResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Events" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.1.3 Sequence diagram

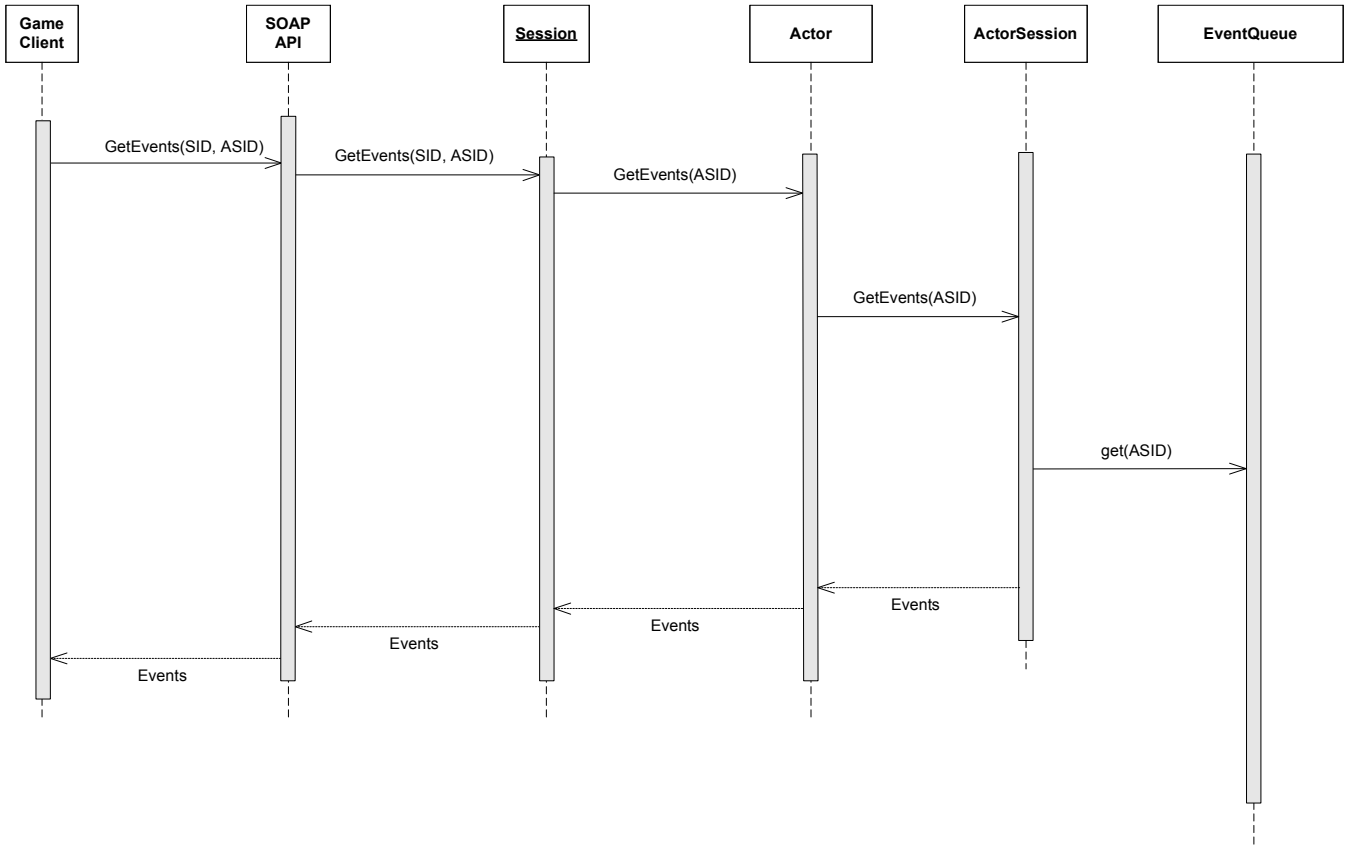

Figure 30: Get Event Operation

6.2.9.2 GetEventHistory

<table>
<tr><td>Name</td><td>GetEventHistory(SID, ASID, Count)</td></tr>
<tr><td>Description</td><td>Retrieves a (maximum) number of events from the event queue in the ActorSession specified by the ASID that have been retrieved before. The maximum number is specified by Count. If more previously retrieved events exist than specified by Count, the most current events are retrieved, i.e. older events are omitted.</td></tr>
<tr><td>Parameter</td><td>
<table>
<tr><th>Key</th><th>Datatype</th><th>Value</th></tr>
<tr><td>SID</td><td>String</td><td>Valid Session ID.</td></tr>
<tr><td>ASID</td><td>String</td><td>ActorSession ID.</td></tr>
<tr><td>Count</td><td>Int</td><td>Number of Events to receive</td></tr>
</table>
</td></tr>
<tr><td>Return Values</td><td>
<table>
<tr><th>Key</th><th>Datatype</th><th>Value</th></tr>
<tr><td>ERR</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr>
<tr><td>DBG</td><td>String</td><td>Error message (optional)</td></tr>
<tr><td>SID</td><td>String</td><td>Valid session id</td></tr>
<tr><td>Events</td><td>Vector</td><td>Vector of Map (*1)</td></tr>
</table>
(*1)
<table>
<tr><th>Key</th><th>Datatype</th><th>Value</th></tr>
<tr><td>Code</td><td>String</td><td>Event Message</td></tr>
<tr><td>ActorNr</td><td>Int</td><td>Actor number who initiated the event</td></tr>
<tr><td>Data</td><td>Anytype</td><td>Event parameters according to code value</td></tr>
</table>
</td></tr>
<tr><td>Related operations/events</td><td>/</td></tr>
<tr><td>Errors</td><td>SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY</td></tr>
</table>

6.2.9.2.1 XSD Complex Type: Parameter

```
<xsd:complexType name="GetEventHistoryRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Count" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.2.2 XSD Complex Type: Return values

```
< <xsd:complexType name="GetEventHistoryResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ASCommandResponse">
      <xsd:sequence>
        <xsd:element ref="tns:Events" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.2.3 Sequence diagram

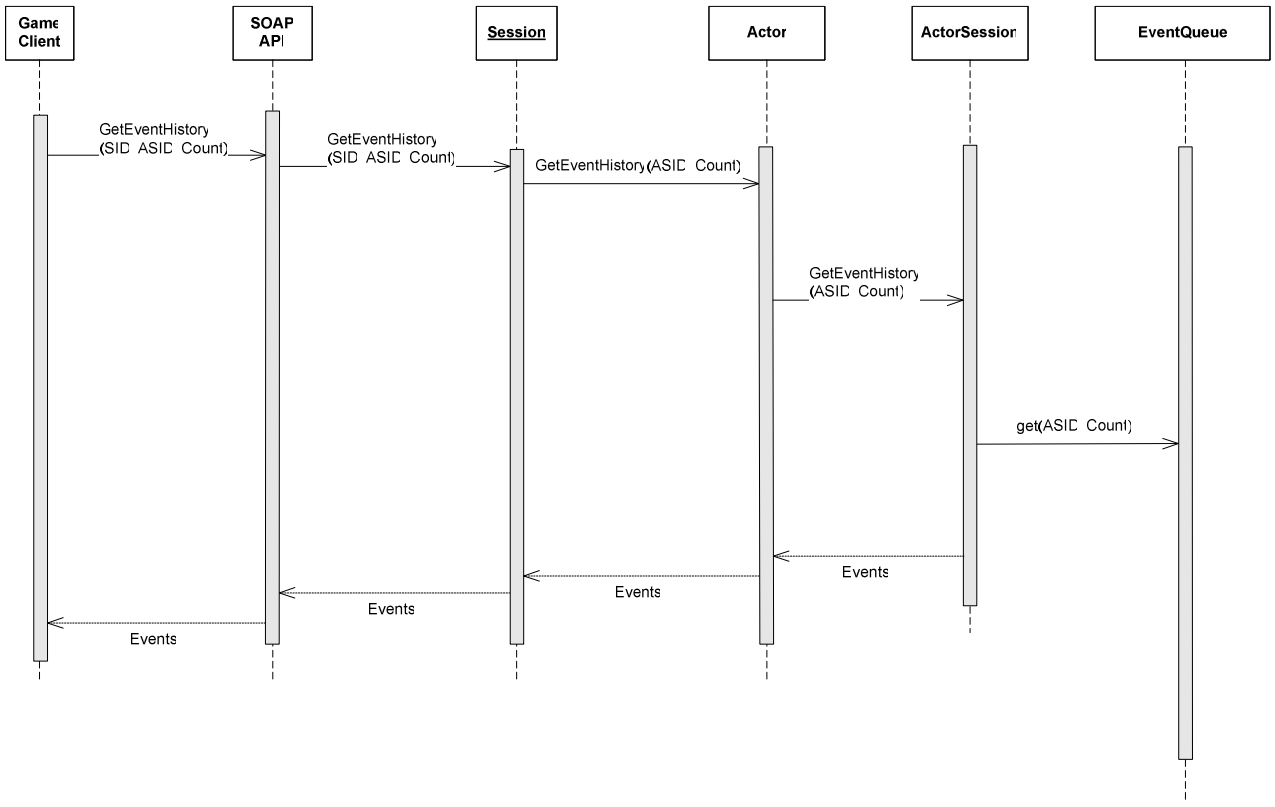

Figure 31: Get event history Operation

6.2.9.3 RaiseEvent

Name	**RaiseEvent**(SID, ASID, Data)
Description	Raises a custom event in the ***ActorSession***.
Parameter	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**SID**</td><td>String</td><td>Valid Session ID.</td></tr><tr><td>**ASID**</td><td>String</td><td>Valid ActorSession ID</td></tr><tr><td>**Data**</td><td>Event</td><td>Event which wants to raise</td></tr></table>
Return Values	<table><tr><th>Key</th><th>Datatype</th><th>Value</th></tr><tr><td>**ERR**</td><td>Int</td><td>Errorcode (0 if no error occurred)</td></tr><tr><td>**DBG**</td><td>String</td><td>Error message (optional)</td></tr><tr><td>**SID**</td><td>String</td><td>Valid session id</td></tr></table>
Related operations/events	
Errors	SESSION_NOT_FOUND, SESSION_EXPIRED, ACCESS_DENIED, ACTORSESSION_NOT_FOUND, ACTORSESSION_EXPIRED, ACTORSESSION_BUSY

6.2.9.3.1 XSD Complex Type: Parameter

```
<xsd:complexType name="RaiseEventRequest">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandRequest">
      <xsd:sequence>
        <xsd:element ref="tns:Data" />
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.3.2 XSD Complex Type: Return values

```
<xsd:complexType name="RaiseEventResponse">
  <xsd:complexContent>
    <xsd:extension base="tns:ActiveASCommandResponse">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

6.2.9.3.3 Sequence diagram

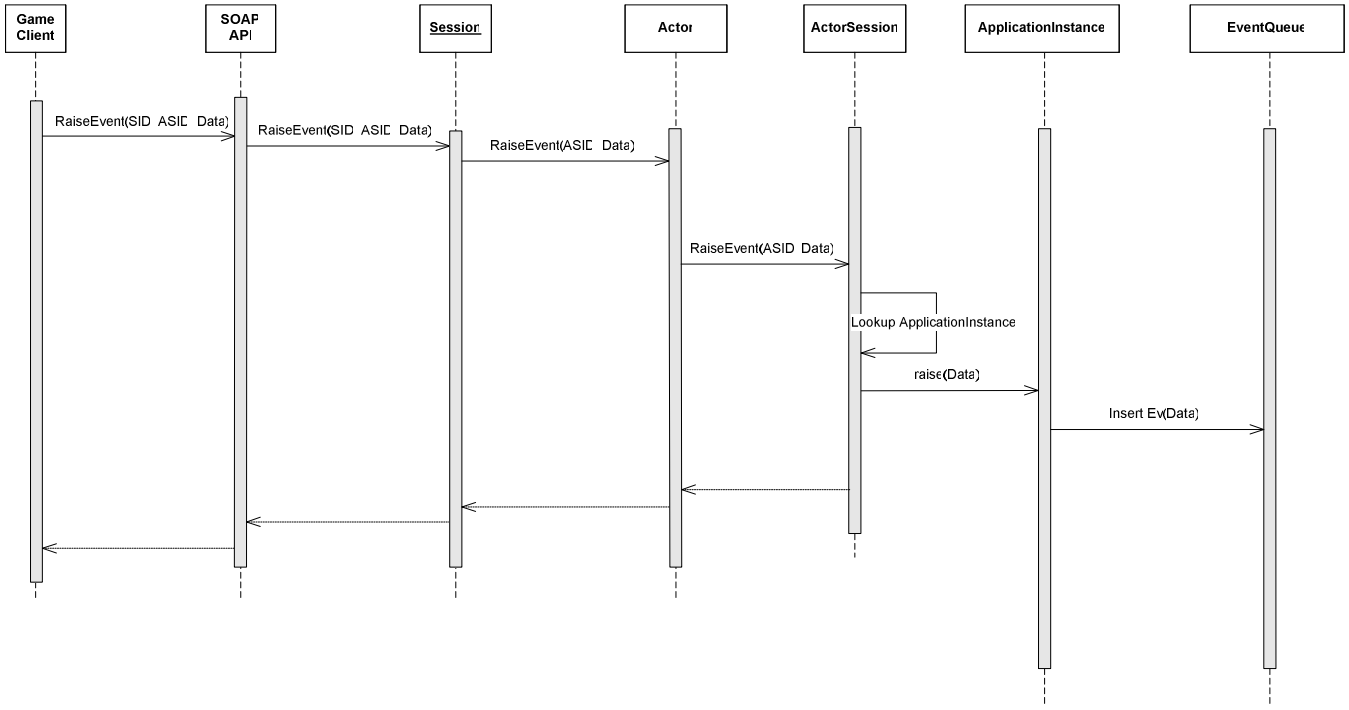

Figure 32: Raise Event Operation

6.3 Events (generated on the server side)

The following section includes tables that states, which command can raise what events and a detailed description of the events. In order for the client to receive events, it has to actively request pending events via the GetEvents() operation.

6.3.1 Events overview

	EvDeactivate	EvActivate	EvJoin	EvStart	EvEnd	EvQuit	EvTurn	EvData	EvTxtMsg	EvStatus
Sessionless										
Register										
Login										X
Session Bound										
CreateAI										
CreateAIPrivate										
DateTime										
ForceAIRnd			X							
GetAS										
GetShadow										
GetInv										
GetScorePos										
GetScoresByID										
GetScoresByPos										
GetScoresByUser										
GetShadow										
Logout										X
Ping										
SetShadow										
SetScore										
ActorSession Bound										

	EvDeactivate	EvActivate	EvJoin	EvStart	EvEnd	EvQuit	EvTurn	EvData	EvTxtMsg	EvStatus
Activate		X								
Deactivate	X									
End					X					
GetEventHistory										
GetEvents										
HandoverTurn							X			
Name										
Quit						X				
RaiseEvent										
SendData								X		
SendTxtMsg									X	
Start				X						

6.3.2 Detailed description of Events

6.3.2.1 EvDeactivate

Event Name	**EvDeactivate**		
Description	An ActorSession has been deactivated. The corresponding ActorNr of the Actor that deactivated its ActorSession is specified in the Event.		
Parameter	Key	Type	Value
	Code	String	EvDeactivate
	ActorNr	Int	Actor number who initiated the event.
Type	Broadcast (to all players)		
Related operations	Deactivate		

6.3.2.2 EvActivate

Event Name	**EvActivate**		
Description	An ActorSession has been activated. The corresponding ActorNr of the Actor that activated its ActorSession is specified in the Event.		
Parameter	Key	Type	Value
	Code	String	EvActivate
	ActorNr	Int	Actor number who initiated the event.
Type	Broadcast (to all players)		
Related operations	/		

6.3.2.3 EvJoin

Event Name	**EvJoin**

Description	This event notifies all Actors connected to a specific ApplicationInstance that another Actor has joined the ApplicationInstance. Actor number und username are given as part of the event parameters.
Parameter	<table><tr><th>Key</th><th>Type</th><th>Value</th></tr><tr><td>**Code**</td><td>String</td><td>EvJoin</td></tr><tr><td>**ActorNr**</td><td>Int</td><td>Actor number who initiated the event.</td></tr><tr><td>**Username**</td><td>String</td><td>Username.</td></tr></table>
Type	Broadcast (to all players)
Related operations	ForceAIRnd

6.3.2.4 EvStart

Event Name	**EvStart**
Description	This event is sent to all players when the team leader uses the Start operation.
Parameter	<table><tr><th>Key</th><th>Type</th><th>Value</th></tr><tr><td>**Code**</td><td>String</td><td>EvStart</td></tr><tr><td>**ActorNr**</td><td>Int</td><td>Actor number who initiated the event.</td></tr></table>
Type	Broadcast (to all players)
Related operations	Start

6.3.2.5 EvEnd

Event Name	**EvEnd**
Description	This event is sent to all players when the ApplicationInstance Master uses the End operation.
Parameter	<table><tr><th>Key</th><th>Type</th><th>Value</th></tr><tr><td>**Code**</td><td>String</td><td>EvEnd</td></tr><tr><td>**ActorNr**</td><td>Int</td><td>Actor number who initiated the event.</td></tr></table>
Type	Broadcast (to all players)
Related operations	End

6.3.2.6 EvQuit

Event Name	**EvQuit**		
Description	This event is sent to all players when a player quits the game.		
Parameter	Key	Type	Value
	Code	String	EvQuit
	ActorNr	Int	Actor number who initiated the event.
Type	Broadcast (to all players)		
Related operations	Quit		

6.3.2.7 EvTurn

Event Name	**EvTurn**		
Description	Notifies a particular player to make his move.		
Parameter	Key	Type	Value
	Code	String	EvTurn
	ActorNr	Int	Actor number who initiated the event.
	Who	Int	Actor number who is in turn.
	Data	Hashtable	Data in turn.
Type	Broadcast (to all players)		
Related operations	HandoverTurn		

6.3.2.8 EvData

Event Name	**EvData**		
Description	Notifies all players connected to a specific ApplicationInstance about new data.		
Parameter	Key	Type	Value
	Code	String	EvData
	ActorNr	Int	Actor number who initiated the event.
	Data	Hashtable	Data.
Type	Broadcast (to all players)		
Related operations	SendData		

6.3.2.9 EvTxtMsg

Event Name	**EvTxtMsg**		
Description	Notifies all players connected to a specific ApplicationInstance about a new text message.		
Parameter	Key	Type	Value
	Code	String	EvTxtMsg
	ActorNr	Int	Actor number who initiated the event.
	Msg	String	Text message.
Type	Broadcast (to all players)		
Related operations	SendTxtMsg		

6.3.2.10 EvStatus

<table>
<tr><td>Event Name</td><td>EvStatus</td></tr>
<tr><td>Description</td><td>Notifies all players connected to a specific ApplicationInstance about a login / logoff operation.</td></tr>
<tr><td>Parameter</td><td>
<table>
<tr><th>Key</th><th>Type</th><th>Value</th></tr>
<tr><td>Code</td><td>String</td><td>EvStatus</td></tr>
<tr><td>ActorNr</td><td>Int</td><td>Actor number who initiated the event.</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Type</td><td>Broadcast (to all players)</td></tr>
<tr><td>Related operations</td><td>Login, Logout</td></tr>
</table>

6.4 Error Handling

The following sections include tables that states, which command can trigger what errors and a detailed description of the errors.

6.4.1 Errors overview

	USER_NOT_REGISTERED	USER_ALREADY_REGISTERED	USER_WRONG_PASSWORD		SESSION_NOT_FOUND	SESSION_EXPIRED	ACCESS_DENIED	ACTORSESSION_NOT_FOUND	ACTORSESSION_EXPIRED	ACTORSESSION_BUSY		OPEN_APPLICATIONINSTANCE_NOT_FOUND	ACTOR_NOT_FOUND		SCORE_TABLE_NOT_FOUND	HIGHSCORE_MAXPOS_EXCEEDED	HIGHSCORE_NOT_FOUND		PARAMETER_OUT_OF_RANGE
Sessionless																			
Register		X																	
Login	X		X										X						
Session Bound																			
CreateAI					X	X	X												X
CreateAIPrivate					X	X	X						X						X
DateTime					X	X	X												
ForceAIRnd					X	X	X					X							X
GetAS					X	X	X												
GetShadow					X	X	X										X		
GetInv					X	X	X												
GetScorePos					X	X	X										X		
GetScoresByID					X	X	X									X	X		
GetScoresByPos					X	X	X								X	X			
GetScoresByUser					X	X	X								X	X	X		

	USER_NOT_REGISTERED	USER_ALREADY_REGISTERED	USER_WRONG_PASSWORD		SESSION_NOT_FOUND	SESSION_EXPIRED	ACCESS_DENIED	ACTORSESSION_NOT_FOUND	ACTORSESSION_EXPIRED	ACTORSESSION_BUSY		OPEN_APPLICATIONINSTANCE_NOT_FOUND	ACTOR_NOT_FOUND		SCORE_TABLE_NOT_FOUND	HIGHSCORE_MAXPOS_EXCEEDED	HIGHSCORE_NOT_FOUND		PARAMETER_OUT_OF_RANGE
GetShadow					X	X	X												
Logout					X	X	X												
Ping					X	X	X												
SetShadow					X	X	X								X				
SetScore					X	X	X								X				
ActorSession Bound																			
Activate					X	X	X	X	X	X									
Deactivate					X	X	X	X	X	X									
End					X	X	X	X	X	X									
GetEventHistory					X	X	X	X	X	X									
GetEvents					X	X	X	X	X	X									
HandoverTurn					X	X	X	X	X	X									
Name					X	X	X	X	X	X									
Quit					X	X	X	X	X	X									
RaiseEvent					X	X	X	X	X	X									
SendData					X	X	X	X	X	X									
SendTxtMsg					X	X	X	X	X	X									
Start					X	X	X	X	X	X									

Table 3: Operations and possible errors

6.4.2 Detailed Description of Errors

6.4.2.1 USER_NOT_REGISTERED

Error Name	USER_NOT_REGISTERED
Description	The user supplied for a login operation is not registered.
Related operations	Login

6.4.2.2 USER_ALREADY_REGISTERED

Error Name	USER_ALREADY_REGISTERED
Description	It was attempted to register with a username that is already registered. The application should suggest choosing another username.
Related operations	Register

6.4.2.3 USER_WRONG_PASSWORD

Error Name	USER_WRONG_PASSWORD
Description	The user supplied an incorrect password for a login operation.
Related operations	Login

6.4.2.4 SESSION_NOT_FOUND

Error Name	SESSION_NOT_FOUND
Description	The session id supplied is not valid.
Related operations	All excluding Login, Register

6.4.2.5 SESSION_EXPIRED

Error Name	SESSION_EXPIRED
Description	The session for the supplied session id is expired.
Related operations	All excluding Login, Register

6.4.2.6 ACCESS_DENIED

Error Name	ACCESS_DENIED
Description	Access to this operation has not been granted by the server.
Related operations	All excluding Login, Register

6.4.2.7 ACTORSESSION_NOT_FOUND

Error Name	ACTORSESSION_NOT_FOUND
Description	The ActorSession id supplied is not valid.
Related operations	All ActorSession bound operations

6.4.2.8 ACTORSESSION_EXPIRED

Error Name	ACTORSESSION_EXPIRED
Description	The ActorSession for the supplied ActorSession id is expired.
Related operations	All ActorSession bound operations

6.4.2.9 ACTORSESSION_BUSY

Error Name	ACTORSESSION_BUSY
Description	The actor session being specified for the operation is already in use in another client/server-session. This might happen if two client devices subsequently attempt to activate the same ActorSession.
Related operations	All ActorSession bound operations

6.4.2.10 OPEN_APPLICATIONINSTANCE_NOT_FOUND

Error Name	OPEN_APPLICATIONINSTANCE_NOT_FOUND
Description	No open application instance found.
Related operations	ForceAIRnd

6.4.2.11 ACTOR_NOT_FOUND

Error Name	ACTOR_NOT_FOUND
Description	The actor specified by the actor id was not found.
Related operations	CreateAIPrivate

6.4.2.12 SCORE_TABLE_NOT_FOUND

Error Name	SCORE_TABLE_NOT_FOUND
Description	The score table for the supplied score table id was not found.
Related operations	SetScore, GetScoresByPos, GetScoresByUser, SetShadow

6.4.2.13 HIGHSCORE_MAXPOS_EXCEEDED

Error Name	HIGHSCORE_MAXPOS_EXCEEDED
Description	One or more scores (specified by the positional parameter and the count parameter) are not inside the valid range of the score table.
Related operations	GetScoresByID, GetScoresByPos, GetScoresByUser

6.4.2.14 HIGHSCORE_NOT_FOUND

Error Name	HIGHSCORE_NOT_FOUND
Description	The score specified by the positional parameter (id, user, absolute position) does not exist in the score table.
Related operations	GetScoresByID, GetScoresByUser, GetScoresByPos, GetShadow

6.4.2.15 PARAMETER_OUT_OF_RANGE

Error Name	PARAMETER_OUT_OF_RANGE
Description	A parameter supplied with the operation is outside its valid range. The server shall specify which parameter is out of range along with tbe actual value given.
Related operations	CreateAIPrivate, ForceAIRnd

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Versions: OMA-TS-Game-Services-Client-Server-Interface-V1_0	06 May 2004	all	- Migrate of specification document "OMA-GS-2004-0012-Gaming-Platform-2.0" to latest OMA Spec Template "OMA-Template-Spec-20040205.doc" - Introduction - Making figures more OMA neutral - Error handling - Overall update
	22 Jun 2004	all	Bangkok meeting walk through. Lots of different changes, mainly editorial.
	24 Aug 2005	all	Montreal meeting walk through. Lots of different changes, mainly editorial.
	12 Dec 2005	Appendix C	Added the UAProf appendix.
	13 Dec 2005	all	Make consistency of document name and put into newest template
	10 Feb 2006	All	Updated according to OMA-CONRR-GS-CSI-1.0-V1_0-20060203-D
Candidate Version: OMA-TS-Game-Services-Client-Server-Interface-V1_0	07 Mar 2006	n/a	Status changed to Candidate by OMA TP: OMA ref# OMA-TP-2006-0089-GS-CSI-V1_0-for-candidate-approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

B.1 SCR for GSCSI Client

Item	Function	Reference	Status	Requirement
GSCSI-LR-C-001	Login	Section 6.2.1.1	M	
GSCSI-LR-C-002	Logout	Section 6.2.1.2	M	
GSCSI-LR-C-003	Registration	Section 6.2.1.3	M	
GSCSI-GPE-C-004	GetAS	Section 6.2.2.1	M	
GSCSI-GPE-C-005	GetInv	Section 6.2.2.2	M	
GSCSI-GPE-C-006	Activate	Section 6.2.2.3	M	
GSCSI-GPE-C-007	Deactivate	Section 6.2.2.4	M	
GSCSI-GPE-C-008	Name	Section 6.2.2.5	M	
GSCSI-GCM-C-009	CreateAI	Section 6.2.3.1	M	
GSCSI-GCM-C-010	CreateAIPrivate	Section 6.2.3.2	M	
GSCSI-GCM-C-011	ForceAIRnd	Section 6.2.3.3	M	
GSCSI-GPL-C-012	Start	Section 6.2.4.1	M	
GSCSI-GPL-C-013	End	Section 6.2.4.2	M	
GSCSI-GPL-C-014	Quit	Section 6.2.4.3	M	
GSCSI-GPL-C-015	Handoverturn	Section 6.2.4.4	M	
GSCSI-GPL-C-016	SendData	Section 6.2.4.5	M	
GSCSI-MSG-C-017	SendTxtMsg	Section 6.2.5.1	M	
GSCSI-HS-C-018	SetScore	Section 6.2.6.1	M	
GSCSI-HS-C-019	GetScorePos	Section 6.2.6.2	M	
GSCSI-HS-C-020	GetScoresByPos	Section 6.2.6.3	M	
GSCSI-HS-C-021	GetScoresByID	Section 6.2.6.4	M	
GSCSI-HS-C-022	GetScoresByUser	Section 6.2.6.5	M	
GSCSI-SH-C-023	SetShadow	Section 6.2.7.1	M	
GSCSI-SH-C-024	GetShadow	Section 6.2.7.2	M	
GSCSI-MI-C-025	Ping	Section 6.2.8.1	M	
GSCSI-MI-C-026	DateTime	Section 6.2.8.2	M	
GSCSI-EQO-C-027	GetEvents	Section 6.2.9.1	M	
GSCSI-EQO-C-028	GetEventHistory	Section 6.2.9.2	M	
GSCSI-EQO-C-029	RaiseEvent	Section 6.2.9.3	M	
GSCSI-EV-C-030	EvDeactivate	Section 6.3.1	M	
GSCSI-EV-C-031	EvActivate	Section 6.3.2.2	M	
GSCSI-EV-C-032	EvJoin	Section 6.3.2.3	M	
GSCSI-EV-C-033	EvStart	Section 6.3.2.4	M	
GSCSI-EV-C-034	EvEnd	Section 6.3.2.5	M	
GSCSI-EV-C-035	EvQuit	Section 6.3.2.6	M	
GSCSI-EV-C-036	EvTurn	Section 6.3.2.7	M	
GSCSI-EV-C-037	EvData	Section 6.3.2.8	M	
GSCSI-EV-C-038	EvTxtMsg	Section 6.3.2.9	M	

B.2 SCR for GSCSI Server

Item	Function	Reference	Status	Requirement
GSCSI-LR-S-001	Login	Section 6.2.1.1	M	
GSCSI-LR-S-002	Logout	Section 6.2.1.2	M	
GSCSI-LR-S-003	Registration	Section 6.2.1.3	M	
GSCSI-GPE-S-004	GetAS	Section 6.2.2.1	M	
GSCSI-GPE-S-005	GetInv	Section 6.2.2.2	M	
GSCSI-GPE-S-006	Activate	Section 6.2.2.3	M	
GSCSI-GPE-S-007	Deactivate	Section 6.2.2.4	M	
GSCSI-GPE-S-008	Name	Section 6.2.2.5	M	
GSCSI-GCM-S-009	CreateAI	Section 6.2.3.1	M	
GSCSI-GCM-S-010	CreateAIPrivate	Section 6.2.3.2	M	
GSCSI-GCM-S-011	ForceAIRnd	Section 6.2.3.3	M	
GSCSI-GPL-S-012	Start	Section 6.2.4.1	M	
GSCSI-GPL-S-013	End	Section 6.2.4.2	M	
GSCSI-GPL-S-014	Quit	Section 6.2.4.3	M	
GSCSI-GPL-S-015	Handoverturn	Section 6.2.4.4	M	
GSCSI-GPL-S-016	SendData	Section 6.2.4.5	M	
GSCSI-MSG-S-017	SendTxtMsg	Section 6.2.5.1	M	
GSCSI-HS-S-018	SetScore	Section 6.2.6.1	M	
GSCSI-HS-S-019	GetScorePos	Section 6.2.6.2	M	
GSCSI-HS-S-020	GetScoresByPos	Section 6.2.6.3	M	
GSCSI-HS-S-021	GetScoresByID	Section 6.2.6.4	M	
GSCSI-HS-S-022	GetScoresByUser	Section 6.2.6.5	M	
GSCSI-SH-S-023	SetShadow	Section 6.2.7.1	M	
GSCSI-SH-S-024	GetShadow	Section 6.2.7.2	M	
GSCSI-MI-S-025	Ping	Section 6.2.8.1	M	
GSCSI-MI-S-026	DateTime	Section 6.2.8.2	M	
GSCSI-EQO-S-027	GetEvents	Section 6.2.9.1	M	
GSCSI-EQO-S-028	GetEventHistory	Section 6.2.9.2	M	
GSCSI-EQO-S-029	RaiseEvent	Section 6.2.9.3	M	
GSCSI-EV-S-030	EvDeactivate	Section 6.3.1	M	
GSCSI-EV-S-031	EvActivate	Section 6.3.2.2	M	
GSCSI-EV-S-032	EvJoin	Section 6.3.2.3	M	
GSCSI-EV-S-033	EvStart	Section 6.3.2.4	M	
GSCSI-EV-S-034	EvEnd	Section 6.3.2.5	M	
GSCSI-EV-S-035	EvQuit	Section 6.3.2.6	M	
GSCSI-EV-S-036	EvTurn	Section 6.3.2.7	M	
GSCSI-EV-S-037	EvData	Section 6.3.2.8	M	
GSCSI-EV-S-038	EvTxtMsg	Section 6.3.2.9	M	

Appendix C. Usage of OMA UAProf with CSI 1.0 (Informative)

The CSI enabler may use the OMA UAProf as a way to retrieve GS handset capabilities.

The usage of the UAProf schema can be found in [UAPROF].

The attributes in **Table 3: "UAProf vocabulary"** are specified in [UAPROF].

Support of the attributes of the UAProf schema in this specification shall not be limited to those listed in **Table 3: "UAProf vocabulary"**. The user profile can support other attributes from [UAPROF].

Vendor
Model
SoftwareNumber
ScreenSize
ScreenSizeChar
ColorCapable
AudioInputEncoder
VideoInputEncoder
PointingResolution
CcppAccept-Language
Keyboard
SupportedBearers

Table 3: "UAProf vocabulary"