A survey on Game Backend Services

Ricardo Queirós
Polytechnic Institute of Porto, Portugal

ABSTRACT

The industry of video games is one of the fastest growing sectors in the worldwide economy. One of the key factors to increase engagement and player retention, was the use of various common game concepts such as leaderboards and achievements. The massive use of this approach and the impressive growth of players led to the concept of gamification as a service, later materialized in Game Backend as a Service (GBaaS). Instead of replicating the implementation of the game features in each version of the game for several platforms, GBaaS adhere to a service oriented architecture providing cross-platform game services that lets you easily integrate popular gaming features such as achievements, leaderboards, remote storage and real-time multiplayer in games. This chapter surveys several GBaaS based on the features they offer and on their openness for the integration with computer programming environments.

Keywords: Web Services, Interoperability, REST, Backend Services, Leaderboards, Achievements, Badges, Google Play Games Services.

INTRODUCTION

The use of game concepts in non-game contexts is an effective way to motivate users. Gamification is currently a key trend in different domains, from marketing to e-learning (Hamari, Koivisto & Sarsa, 2014). The massive use of this approach led to the concept of gamification as a service, provided by major players such as Google and Microsoft. These services leverage on their large user base to provide support for game progress mechanics such as points, leaderboards and badges.

The games industry is growing exponentially in the worldwide economy (Zackariasson & Wilson, 2012). According to the research company Gartner, global video game sales will reach $111.1 billion in 2015, due in part to the growth in mobile game play and the recent release of the new generation of game consoles. In order to increase engagement and player retention, video games include several common features such as leaderboards and achievements. The massive use of this approach and the impressive growth of players led to the concept of gamification as a service, later materialized in **Game Backend as a Service (GBaaS)**. The approach is simple. Instead of replicating the implementation of the game features in each version of the game for various platforms, GBaaS adhere to a service oriented architecture providing cross-platform game services that lets you easily integrate popular gaming features such as achievements, leaderboards, remote storage and real-time multiplayer in mobile games.
Focusing on e-learning, the integration of game concepts in learning environments helps students to remain focused and to fulfil their course goals (Burguillo, 2010) (Siddiqui, Khan & Akhtar, 2008). However, the implementation of gamification in these domains is often trapped in ad-hoc solutions or supported by specific platforms (for instance, the badges in Moodle), instead of using approaches such as those provided by GBaaS. In the following subsections we briefly summarize the main common game features that can be

applied to the teaching-learning process. Then, we compare six GBaaS regarding social and technical features. This study is part of an effort to select an GBaaS on which to base the development of a service for gamification of learning activities.

GAME CONCEPTS

Games are more interesting when players are able to achieve goals and compete against other players. These features foster retention and competitiveness, and are applicable also in the gamification of e-learning activities. The following list enumerates the most common game concepts:

- **Leaderboards:** databases that keep scores. They allow users to post their scores in a game and compare themselves with other players' scores. They measure the success of a player in a game.
- **Achievements:** goals/challenges set in a game that players managed to accomplish. Achievements give players a motivation to keep playing, to earn as many as possible, and a way compare themselves with other players. The fulfilment of a goal may enhance the status of the player or unlock access to other levels, for instance.
- **Multiplayer:** is a play mode that allows several players to simultaneously cooperate or compete in a game. This feature supports a range of other subfeatures, such as challenges, where players compete each other on either a score challenge or an achievement challenge, and matchmaking games for real-time, turn-based, or self-hosted matches.
- **Saved games:** allow the remote storage (in the cloud) of game data, for instance, the state and the players progress in the game.
- **Quests:** are periodic game challenges that players can complete to earn rewards. This way, developers can launch periodic challenges to their gaming communities.
- **Gifts:** allow players to send/request game resources or items to/from friends (for instance, in their Google+ circles).
- **Matchmaking:** automatically sets up game matches and finds opponents based on parameters set by the game developer. Usually only a specific number of players can be matched at the same time.

GAME BACKEND SERVICES

A **Backend-as-a-service (BaaS)** is a cloud computing service model acting as a middleware component that allows developers to connect their Web and mobile applications to cloud services via application programming interfaces (API) and software developers' kits (SDK). BaaS features include cloud storage, push notifications, server code, user and file management, social networking integration, location services, and user management as well as many other backend services. These services have their own API, allowing them to be integrated into applications in fairly simple way.

A **Game-Backend-as-a-Service (GBaaS)** is a subset of a BaaS that includes cross-platform solutions for the typical game concepts identified in the previous subsection. During the development process of a game (or a generic application) developers must choose between building their own back-end services or using an available game back-end platform. This last option is usually preferred since GBaaS include several services specifically tailored for game development. These services allow developers to focus on the game logic by freeing them from implementing boiler plate features.

Figure 1. Game Backend Services

Typically, a GBaaS includes several gamification features which frees the programmer to the creation of good games rather than reinventing the wheel with the implementation of common gaming features such as leaderboards and achievements.

Most of the GBaaS on the Web offer a set of gaming features through a uniform API and with SDKs in several flavors. One of the features that distinguish GBaaS is regarding its pricing strategy: free or freemium. The former allows the programmer with a previous registration to access all the features of the GBaaS. The later is a pricing strategy by which a product or service (typically a game or a web service) is provided free of charge, but money (premium) is charged for proprietary features, functionality, or virtual goods.

Cloud service providers ranging from Infrastructure providers (IaaS), platform providers (PaaS) to software service providers (SaaS) – use freemium pattern in their business models.

SURVEY

In this section several GBaaS are compared according to their social and technical features. Given the number of GBaaS found (32) it would be impracticable to study them all. Therefore, four of the most popular GBaaS are detailed (Google Play Game Services, Yahoo Backend Game Service, GameUp, GameSparks) and six are used for the social and technical features comparison.

Google Play Game Services

Google Play Games Services (GPGS) (Hartrell, 2015) is an online gaming service, introduced at the Google I/O 2013 Developer Conference, and managed by Google. It features several gaming capabilities such as real-time multiplayer, cloud saves, social and public leaderboards, achievements, and anti-piracy features. The service allows developers to include features in their games without having to develop those features themselves.

Programmers should use the Google Play Developer Console to manage games services and configure metadata for authorizing and authenticating in the game. Then, start integrating popular gaming features in mobile and Web games by using the Google Play games services APIs for five flavors: Android, iOS, C++, REST and Unity. This BaaS has several services that should be introduced, more precisely:

- **Saved games:** gives a convenient way to save players' game progression to Google's servers. The game can retrieve the saved game data to allow returning players to continue a game at their last save point from any device.
- **Events and Quests:**
    - **The Events service** allows you to collect cumulative data generated by your players during gameplay and store them in Google's servers for game analytics. You can flexibly define what player data your game should collect; this might include metrics such as how often: players use a particular item, players reaching at a certain level or even players perform some specific game action
    - **The Quests service** complements the events service by allowing you to introduce new time-bound challenges (or quests) that are based on events data. Quests allow you to re-

engage players and incentivize them with some in-game reward or benefit if they succeed, without having to republish your entire game in the Google Play Developer Console.

- **Multiplayer:** connect multiple players together in a single game session and transfer data messages between connected players. Using the real-time multiplayer API can help to simplify your game development effort because the API handles several taskas, such as:
  - Manages network connections to create and maintain a real-time multiplayer room;
  - Provides a player selection user interface (UI) to invite players to join a room;
  - Stores participant and room state information on the Google Play games services servers;
  - Sends room invitations and updates to players.

  GPGS also supports **turn-based multiplayer games**, where a single shared state is passed between multiple players, and only one player has permission to modify the shared state at a time. Players take turns asynchronously according to an order of play determined by the game.
- **Leaderboards and achievements:** The GPGS Publishing API allows you to automate frequent tasks having to do with games production and distribution, such as, creating and modifying achievement and leaderboard listings (Figure 2).

The GPGS also includes the **Management API** that allows the issue of REST calls to programmatically control the metadata underlying the GPGS features. This API simplifies your features testing and gives flexibility when managing your games to address cheating and fix player accounts. With the Management API, it's possible to reset values for leaderboards and achievements and even hide players from leaderboards in published games.

Figure 2. Google Play Game Services leaderboard

The REST API for Google Play games services allows you to issue REST calls to set and retrieve game services data from your web game. There are two ways you can communicate with Google Play games services:

- In a client-only setup, your game only communicates with Google Play games services through the browser using JavaScript.
- In a client-server situation, your game typically communicates with Google Play games services from your server using a server-side language such as Java, Ruby, or PHP

The following code shows how to get leaderboard data from the GPGS:

```
GET https://www.googleapis.com/games/v1/leaderboards
```

If successful, this method returns a JSON response body with the following structure:

```
{
  "kind": "games#leaderboardListResponse",
  "nextPageToken": string,
  "items": [
    leaderboards Resource
  ]
}
```

The body includes three properties: the `kind` property uniquely identifies the type of this resource. Value is always the fixed string `games#leaderboardListResponse`. The `nextPageToken` defines the token corresponding to the next page of results. Finally, the `items` property references a list with the actual items of the leaderboard.

Yahoo Backend Game Service

The Yahoo Backend Game Service (YBGS)[1] is a powerful BaaS where a programmer can power his game on Yahoo Games Network without any server management. The BaaS has a set of services to build and manage the game, namely:

- **Achievements:** is a service with a series of API calls that allows you to reward players when a milestone in a game is reached;
- **Saved data:** to enable development of infinitely scalable games, Yahoo Games Network provides BigDB; a database solution built especially for games, that scales transparently with any amount of usage and load;
- **Error logs:** makes it easy to identify and fix errors you otherwise wouldn´t know about by providing a centralized location where information about runtime errors encountered in your game is saved;
- **GameFS:** is a hosted file system for your games. You can store anything in GameFS that needs to be distributed to your players - SWF files, images, levels, graphics, html pages, etc.;
- **Game requests:** is a unified system for sending requests from a user in your game, to other users in the game, or to people who aren't playing the game yet, and for handling received requests when the receiver is playing or starts playing the game;
- **Multiplayer:** allows you to build real-time games with your own custom server-side code, and to run that code on our dedicated and distributed game servers;
- **Notifications:** Push Notifications enable mobile application providers to push data to the client, as opposed to requiring clients to poll for updates;
- **Pay vault:** makes it easy to add ingame payments and/or a complete virtual currency system to your game, complete with an optional secure bought-item management system;
- **Player insight:** is a comprehensive analytics tool that comes automatically when you use the Backend Services of the Yahoo Games Network with your game;
- **Quick connect:** makes it easy to securely connect and authenticate towards the Backend Services with users from third party user databases, like Kongregate and Facebook;
- **Site box**: is the easiest way to build and host sites and Facebook apps for your games.

In order to use the YBGS it is andatory to download the SDK that comes in five flavors: ActionScript 3, Unity3D, C#, Andoid or iOS.

Firstly, to use any of the Backend Services, your game client must first identify itself and the current player by using the Authenticate method of the Client Library you're using. There are several types of authentication from basic to using the Facebook.

After the authentication, it is possible to use the above services. This code example in C# shows how to send a game request using the API.

```
//Send a game request
client.GameRequests.Send(
    "gift",                                    //type
```

[1] Yahoo Games network. Retrieved February 15, 2016, from https://gamesnet.yahoo.net/documentation/services/.

```
    new Dictionary<string, string>(){            //payload
        { "content", "cow" },
        { "source", "dailygift" },
    },
    "otherUser...");                             //recipient

//Refresh waiting requests
client.GameRequests.Refresh(delegate() {
    //Check how many requests are waiting
    var waiting = client.GameRequests.WaitingRequests.Length;

    //Go over all requests
    foreach(GameRequest request in client.GameRequests.WaitingRequests) {
        if (request.Type == "invite") {
            //Do something with invite requests...
        } else {
            //Handle other requests...
        }

        //Delete the request after handling it
        client.GameRequests.Delete(request);
    }
});
```

GameUp

GameUp[2] is a scalable and fast gaming service for game developers. The service provides the features and functionality provided by game servers today. Its main goal is to enable game developers to focus on being as creative as possible and building great games. By leveraging the GameUp service it is easy to add social login, gamer profiles, cloud game storage, and many other features.

The authentication service uses HTTPS with Basic Authentication to enforce access controls to the Game API. The service also provides a simple mechanism for your game to implement social login, allowing gamers to sign in using Facebook, Google, or Tango, and start playing. The main features are:

- **Leaderboards:** way to add a social and competitive element to any game. They're a fun way to drive competition among your players. The service supports an unlimited number of individual leaderboards per game; each one is a standalone scoreboard which tracks separate values. Heroic Labs has no special requirement on what the score value should represent from your game. A leaderboard is created with a sort order which will order rankings. If you're using lap time or currency as a score value you'll want to order the results in ASC or DESC mode as required. You can use a leaderboard to track any score you like! Some good examples are: Highest Points, Longest Survival Time, Fastest Lap Time, Quickest Level Completion, and anything else which is worth competing over in your game;
- **Achievements:** a way to reward players, encourage exploration, and increase replayability. Every achievement awards a set number of points, between 10 and 100, up to a total of 1000 points per game. This will contribute to a user's overall profile score, adding an extra incentive to both continue playing and play new games. There are two types of achievements in the service: normal and incremental; and all achievements can be created with one of three possible states: visible, secret, or hidden;

[2] Heroic Labs. Retrieved February 15, 2016, from https://heroiclabs.com/.

- **Shared storage:** enables any player to find and interact with shared data and content created by any other player. Users create any number of Shared Storage JSON objects that can be found and retrieved using our powerful query engine;
- **Multiplayer:** brings communities together to interact and compete in new and exciting ways. The service allows gamers to either challenge a set of given players or instantly find opponents anywhere in the world, and jump right into the action;
- **Push notifications:** allows you to notify a group of gamers about new content, offers, events, and more. It's a superior option to long-polling processes as it requires much less power and bandwidth, both of which are in relatively short supply on mobile devices.

The service uses HTTPS and JSON as the protocol for the Game API. All social and competitive features are fully cross-device. The service has been developed with "edge" connectivity in mind. It has first-class mobile support with official SDKs for Unity, Android, iOS, and HTML5/FirefoxOS. The service has full support for CORS negotiation; a pre-flight request used by browsers to negotiate permissions for cross-domain requests. This enables the domain host of a game to make AJAX requests to the game backend service.

As the code example, it is presented the way to display all achievements for a gamer including unlocked achievements and progress so far. For that you will need a gamer token as well as an API Key. In order to display the achievements unlocked (and in progress) for a gamer:

```
curl -X GET https://api.heroiclabs.com/v0/gamer/achievement \
        --user "{YourAPIKey}:{YourGamerToken}"
```

GameSparks

GameSparks[3] is a cloud-based platform that provides developers with the tools they need to build the server-side components of their games and then manage them as a service post launch. The GameSparks API allows developers to access and integrate the functionality of GameSparks with other applications and to create new applications. Some example API methods include managing social media, managing in-app purchases, and managing engagement.

The GBaaS uses HTTP, JSON and REST as the protocol/formats for the remote communication from clients to the game servers. Game Sparks divides its set of features in four facets:

- **Core features:** main features of the backend
    - Authentication – allows Out-of-the-box integrations with Facebook, Twitter, Google, Apple, Microsoft, Amazon and Steam enable social features, in-app purchases and push notifications.
    - Push notifications - Manage all of the communication with the players from one backend.
    - Binary asset management - allow the management of the games binary assets from the server side.
    - Cloud data - Create and manage rich server-side logic using our online cloud coding tool and debugger. Add logic to existing features using our extensive event library or build you own server-side API to add any server-side capability to your game. Get full access to your game's underlying data via the NoSQL Explorer Interface.
- **Optimisation and Management features:** allows the fine tuning it in response to player action and maximising engagement and retention
    - Player management - Full access to player history and summaries of activity. View and edit player records and reward top players or remove unwanted players.

[3] GameSparks. Retrieved February 15, 2016, from http://www.gamesparks.com/.

  - Segmentation - Group players into segments, analyse their behaviours, and operate specific engagement strategies and game configurations for each.
  - Analytics and Reporting - Store every action a player takes on the backend in its lowest granular form.
- **Social features:** teams, leaderboards, realtime and turn-based multiplayer, chat and social network integration (e.g. Facebook, Twitter, Google+);
- **Economies features:** currencies, virtual goodies, in-app purchases, rewards & achievements.

In-app purchases are purchases made from within a mobile application. Users typically make an in-app purchase in order to access special content or features in an app such as power-ups, restricted levels, virtual money, special characters, boosts, etc. In GBaaS GameSparks, when a player buys something on one platform, they should have it in their inventory when they play on another. Our in-app purchase management is completely cross platform and integrates with the Android, Apple and Windows Phone App Stores out of the box.

GameSparks also includes a set of SDKs to stimulate its use, more precisely, Unity, Marmalade, Unreal Engine 4, Cocos2d, Javascript/HTML, Actionscript and C++. In the case you already have a backend component, the GBaaS supports server-to-server patterns as well through a server-to-server API.

GameSparks also have a game REST API that allows you to retrieve, create or update the full configuration for your game in JSON format. One feature tat distinguishes GameSparks from the others GBaaS is the capability to manage game data (NoSQL data) via a REST interface, this is useful to allow you to control your game data via your own software. For instance, to find data use the following URL:

```
POST: /rest/games/{gameApiKey}/mongo/{stage}/{collection}/find
```

Where `gameApiKey` is the API key of you game and can be found on the overview page of the Portal Configurator; `stage` indicates which database stage, preview or live, to use; and `collection` indicates which collection to perform the action against. Example of the response:

```
[ {
 "_id": {
 "$oid": "536ce4aa4566340888d2d969"
 },
 "_class": "com.gamesparks.common.domain.Player",
 "gameId": 161,
 "deviceRegIds": {},
 "goods": {},
 "achievements": [],
 "currency1": 1,
 "reservedCurrency1": {},
"authTokens": [
 "fda60fce-0ee8-4efb-beb6-ffd99b459bbe"],
 "userName": "gamedude",
 "displayName": "G-D",
 "password": "***************************",
 "online": true,
 "scriptData": {},
 "privateData": {},
 "externalAuthentications": {},
 "externalIds": {},
 "created": 1399645354792
 }
]
```

Comparision of Social and Technical game features

In this section is compared several GBaaS according to their social and technical features. These features are summarized in Table 1.

Table 1. Social and Technical game features

Types	Features	Google	Yahoo	GameUp	GSparks	PlayFab	Photon
Social	Leaderboards	YES	NO	YES	YES	YES	YES
	Achievements	YES	YES	YES	YES	NO	NO
	Multiplayer	YES	YES	NO	YES	YES	YES
	Save data	YES	YES	YES	YES	YES	YES
	Quests	YES	NO	NO	YES	NO	YES
	Gifts	YES	YES	NO	YES	NO	YES
	Matchmaking	NO	NO	YES	NO	YES	YES
Technical	Pricing	Free	Free	Freemium	Freemium	Freemium	Freemium
	Authentication Authorization	Google+	Yahoo Facebook	Facebook	Facebook Twitter	Facebook	Facebook
	Storage	-	BigDB	-	MongoDB	-	-
	Web Service	REST	-	REST	REST	-	REST
	Response format	JSON	-	JSON	JSON	-	JSON
	Platforms	Android iOS C++	ActionScript iOS Android C# Unity	Android iOS Unity	ActionScript C++ Cocos2D JavaScript Marmalade Unity	Android iOS Unity	Android .NET Unity

Social Game Features

The studied GBaaS provide developers with social game services accessed through cross-platform API. These features make the gameplay more competitive and collaborative, and improve social engagement.
Analysing Table 1 one concludes that almost all GBaaS supports leaderboards, multiplayer game mode and cloud storage. Other features such quests and matchmaking are not yet widely supported, probably due to their novelty.

Technical Game Features

The studied GBaaS offer cloud services through API and SDK to various platforms. Regarding princing strategies the GBaaS studied are organized in free or freemium strategies. Regarding authentication almost all GBaaS use the same strategy. Before the game can make any calls to the game services, it must first establish an asynchronous connection with the backend servers and authenticate within the game services. Some GBaaS requires that the players have an account on specific backends (GPGS requires that users have a Google account). Others, such as GameSparks, provides a simple mechanism that allows games to implement social login without any additional code, allowing gamers, for instance, to sign in using a Facebook or Twitter account, and start playing.
The majority of the GBaaS provides a HTTP RESTful API. The format of the data in all HTTP store operations (PUT and POST) are required to be valid JSON. All response data from the GBaaS comes back also in JSON format.

Regarding the REST API reference, the author opinion is that GPGS is the most complete and better documented API. In complement to the REST API most GBaaS support also mobiles. There are examples of SDKs for Android, iOS, and even FirefoxOS (GameUp) mobile native apps. Game engines are also supported and most GBaaS offer SDKs for major game engines such as Unity, and also for cross-platform game development tools such as Marmalade and Cocos2d.

FUTURE RESEARCH DIRECTIONS

The gamification is the application of game-design elements and game principles in non-game contexts. Gamification techniques fosters people's natural desires for socializing, learning, mastery, competition, achievement and status. Thus, the potential of using gamification in other domains, especially complex domains (such as computer programming education), is huge. This is still a novel approach, but examples of this approach are appearing. That is the case of Odin (Paiva, Leal & Queirós, 2015), a gamification service that uses gamification in computer programming education.

CONCLUSION

Games industry is evolving at an unbridled level. This was one of the reasons for the appearance of Game Backend services. The goal of these cloud services is to free the programmer of the implementation of the game infrastructure and give him more time to dedicate in the game logic. The idea is to not replicate the implementation of the game features in each version of the game for several platforms and adhere to a service oriented architecture providing cross-platform game services that lets you easily integrate popular gaming features such as achievements, leaderboards, remote storage and real-time multiplayer in games.

This chapter surveys several GBaaS based on their technical and social features. In general, all the GBaaS studied provide a set of advanced social features that fosters the use of gaming cloud services. Some distinguish points are related with technical points such as the pricing strategy or the platforms supported. Regarding their openness the potential use of them for the integration with complex domains such as, the computer programming domain is enormous. This study is part of an effort to select an GBaaS on which to base the development of a service for gamification of learning activities.

REFERENCES

Burguillo, J.C. (2010). Using game theory and competition-based learning to stimulate student motivation and performance. *Computer Education*. 55(2), 566–575. Retrieved from: http://dx.doi.org/10.1016/j.compedu.2010.02.018.

Hamari, J., Koivisto, J., Sarsa, H. (2014). Does gamification work?–a literature review of empirical studies on gamification in *System Sciences (HICSS), 47th Hawaii International Conference* on. pp. 3025–3034, IEEE.

Hartrell, G. (2015). Unlocking the Power of Google for Your Games. Retrieved from http://android-developers.blogspot.pt/2014/03/unlocking-power-of-google-for-your_17.html.

Paiva, J.C.; Leal, J.P. & Queirós, R. (2015). Odin: A Service for Gamification of Learning Activities, in *Applications and Technologies - 4th International Symposium, SLATE 2015, Madrid, Spain*, *Communications in Computer and Information Science, Springer International Publishing*, ISBN: 978-3-319-27652-6 (print) 978-3-319-27653-3 (online), DOI: 10.1007/978-3-319-27653-3_18.

Siddiqui, A., Khan, M., Akhtar, S. (2008). Supply chain simulator: A scenario-based educational tool to enhance student learning. *Computer Education*. 51(1), 252–261. Retrieved from: http://dx.doi.org/10.1016/j.compedu.2007.05.008.

Zackariasson, P., Wilson, T. (2012). The Video Game Industry: Formation, Present State, and Future. *Taylor & Francis*. Retrieved from: http://books.google.pt/books?id=lgiQNdcDOwC.

KEY TERMS AND DEFINITIONS

Achievement: an award for completing a particular task or meeting an objective in a game.

Application Programming Interface (API): a set of uniform programming interfaces for building software and applications.

Backend As a Service (BaaS): cloud computing service model acting as a middleware component that allows developers to connect their applications to cloud services via API and SDK.

Freemium: pricing strategy with free and paid resources.

Gamification: conceptual model that aims to bring game mechanics to non-game contexts.

Game Backend As a Service (GBaaS): is a subset of a BaaS that includes cross-platform solutions for the typical game concepts.

Leaderboard: a simple board showing the leaders users ranking in a competition.

Software Development Kit (SDK): a set of software development tools that allows the creation of applications for a certain operating system.