



Web Services for the Management of Persistent Online Game Factions

Deliege, Francois; Zimanyi, Esteban

Published in:
2nd Int. Conf. on Web Information Systems and Technologies

Publication date:
2006

Document Version
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Deliege, F., & Zimanyi, E. (2006). Web Services for the Management of Persistent Online Game Factions. In 2nd Int. Conf. on Web Information Systems and Technologies (pp. 441-446). Institute for Systems and Technologies of Information, Control and Communication.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

WEB SERVICES FOR THE MANAGEMENT OF PERSISTENT ONLINE GAME FACTIONS

François Delière

Aalborg University

Fredrik Bajers Vej 5, 9100 Aalborg, Denmark

Email: fdeliege@cs.aau.dk

Esteban Zimányi

Université Libre de Bruxelles

avenue F.D. Roosevelt 50, 1050 Brussels, Belgium

Email: ezimanyi@ulb.ac.be

Keywords: On-line role playing, multi-player game, web services, communication models, community acceptance.

Abstract: Star Wars Combine is a game involving thousands of players in a virtual world. Each player impersonates a character that continues evolving even when the player is not connected. Players have formed groups, called factions, that are self-organized. The more members a faction has, the more complicated to manage it becomes. The goal of this work is to create an infrastructure to allow faction management tools to automatically update their data with the information maintained on the game server. The web services technology is chosen for the various advantages it offers. While this technology is already widely accepted in the business world, its use in a game context is totally new. It seems that web services have gained their place in the game world and will continue to be developed. Hopefully, this experiment will convince other game designers to adapt their platforms in a similar manner.

1 INTRODUCTION

Web services are widely used by several companies to maximize their e-commerce products. In view of all the advantages that such services offer, we wondered whether they could also be usefully integrated into Massively Multi-Player On-line Role-Playing Games, briefly MMORPGs. We tried to apply them to a well-known MMORPG called Star Wars Combine¹, briefly SWC. The experience revealed soon to be fully successful, as we will explain in this paper.

SWC is defined by its main developers as follows :

“The Star Wars Combine is a free massively multi-player on-line role-playing simulation game, based on the Star Wars universe, developed by amateurs during their spare time.”

Through its simulation aspects, SWC introduces and develops the concept of factions. Factions are groups of players who are given in-game tools to organize and manage themselves. The faction leaders have access to various data concerning their factions: membership statistics, reports, inventory status, maps,

etc. With time, a growing need emerged for the factions to be able to retrieve information in a usable format and to import it in their own application.

The project consisted in trying to develop a client-server architecture using web services technology. By invoking remote methods, factions would be able to perform various actions without using the game interface anymore, but by using their own application instead. Among the system specifications, the communication system to be created would have to be easy for both server and client programmers to implement, while offering security features to allow users to authenticate themselves and retrieve the data in relation to their characters' access levels.

This paper is organized as follows. Section 2 briefly presents the SWC game concepts and its current architecture. Section 3 defines our objectives and explains why the web services technology was chosen to set up a new communication infrastructure. Section 4 presents the implementation design. Section 5 discusses security mechanisms such as the authentication and the authorization processes. Section 6 shows how the web services were perceived by the community, as illustrated by some usage statistics. Section 7 concludes with an evaluation of the work and perspectives for future developments.

¹<http://www.swcombine.com/>

2 SWC OVERVIEW

As already mentioned in the introduction, SWC is a MMORPG based on the Star Wars universe, developed by amateurs during their spare time. It offers one of the best Star Wars universe simulation with its own various game mechanisms related to the physics, economics, politics or engineering in the universe.

As in any role-playing game, the character is one of the most important game concepts. Any player joining the game is invited to create his own character with various specific skills that he will be able to enhance throughout the game. In most MMORPGs when a player disconnects, his character is saved in the game database and his avatar disappears from the game. When the player returns, his character reappears somewhere in the game universe. SWC has adopted a different approach: once a player has created a character, his character remains in the game universe until it gets killed. This is referred as characters persistence.

Factions are another main concept of SWC, as shown in Figure 1. Factions are groups of players who are organizing and managing themselves. SWC proposes a complete system allowing any player to create his own faction whenever he wishes. He will, however, have to fulfill a few in-game conditions such as a determined capital, in-game cash, or in-game assets as well as out-of-game conditions such as the creation of a web site. Upon foundation of a new faction, the player may again decide in which area his faction will work and develop, e.g., a faction may specialize in bounty hunting, mining, trading, or any other activities. In order to manage themselves, factions can use tools made available to them through the game interface. Those tools are very useful and offer various ways to manage the faction, but the possibilities to export the data remain limited. While exporting data does not seem very useful for small factions, it is a very attractive feature for most of the medium to large scale factions, that have most of the time developed a dynamic web site as well their own application and tools.

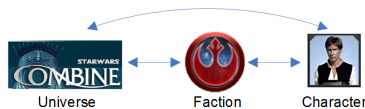


Figure 1: SWC main concepts

Today, SWC counts more than 2000 active players, all interacting with the game engine through a web-based game interface. This Graphical User Interface, briefly GUI, is the only existing way to interact with the game engine.

3 OBJECTIVES AND MOTIVATION

The purpose of creating a web services infrastructure is to allow communications between heterogeneous machines and the game engine. In large factions, groups of players tend to develop their own tools to improve their management capacities and enlarge their game experience. Web services would allow those tools to be automatically updated with fresh data. The three specific objectives of this work are therefore the following:

1. To develop a web services architecture for the game server already in place. By invoking remote methods, factions will be able to perform various actions without using the game interface anymore, but by using their own application instead. Among the system specifications, the communication system to be created will have to be easy for both server and client programmers to implement, while offering security features to allow users to authenticate themselves and retrieve the data in relation to their character access levels.
2. To integrate the web service infrastructure into the already deployed game server with respect of both server security and game rules.
3. To demonstrate the technology usability and measure its acceptance among the whole community.

The decision of using web services technologies to enable remote faction management was made for the following reasons:

- Faction communication has clearly a distributed computing nature: parts of the solution exist in multiple network endpoints.
- The solution has to be built and run by various organizations involving different administration teams.
- The data need to be available to more than just the core application that generates and maintains them.
- Most of the transactions should become automated and therefore extensive use of machine-to-machine communication is preferable.
- The solution needs to be flexible to respond to the various changes that future developments could require from any partner.

While web services have been adopted by major companies such as Google or Amazon, it does not seem that any game has already truly made the step of widely using them. Most existing games using web services as part of their core features are very basic ones such as Tic-Tac-Toe games. Other games are using web services: for example, YaYa LLC² has

²<http://www.yaya.com/>

developed a platform using web services capabilities that allow software designers to create games capturing data on a player's preferences, e.g., for car models. This information is then delivered to a Customer Relationship Management system, e.g., the car manufacturer.

4 DESIGN CHOICES

The SWC technology is an on-line browser-based game model. The server runs the simulation logic and generates web pages to interface with the players. The whole game is therefore centralized in one location and no information is transmitted to third parties' web sites. The primary purpose of the SWC web services is to enable client interaction with the simulation server. Two models of communication exist.

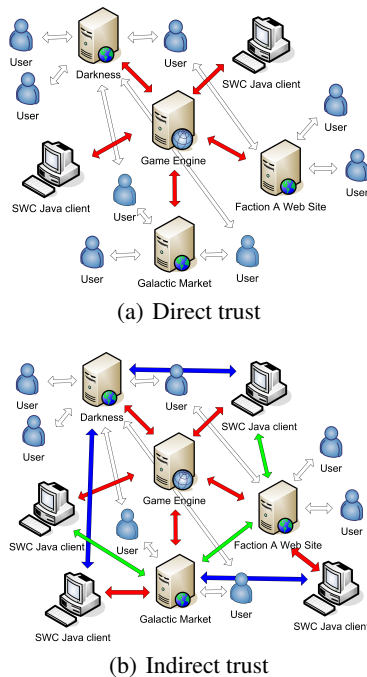


Figure 2: Trust models

The direct trust relationship model, Figure 2(a) enables third parties' applications to directly connect to the central application and perform actions or updates. The GUI becomes one of the many applications that connect to the server and permits user interaction.

In the indirect trust relationship model, Figure 2(b), third parties' clients are enable to interact with each other directly and to develop their own client server architecture that allows client applications to directly interact with the game server.

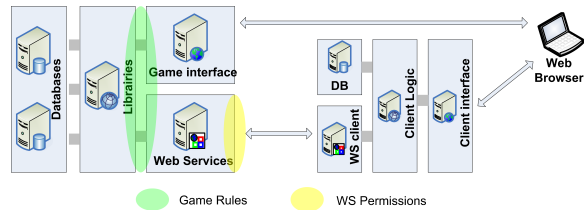


Figure 3: SWC layers web services

Figure 3 shows how the web services infrastructure components (remote objects and remote methods interface, web services logic and management interface) will be integrated into the already existing production server. Web services represent an alternative to the classical web interface used to interact with the game server. They lay between the client and the libraries, next to the web interface. A new set of rules, proper to web services, are created to handle web services access. These authorization rules are completely independent from the "In-Game" rules that belong to the simulation.

5 SECURITY FEATURES

The authentication and authorization mechanisms are not independent from each other. The partners which are involved in the authorization process need to have their identity verified by the authentication process. The SWC security requirements are as follows:

- The administrators need to be able to control which faction has access to a particular web service.
- The game requires a user authentication to perform character actions.
- Factions can not be trusted and should not handle critical user information such as user password.
- A web site can be used by multiple factions.
- A faction can be using multiple web sites.

5.1 Authentication

The web service authentication and authorization processes are composed of three following steps pictured in Figure 4:

1. **User authentication and creation of a authentication contract:** When a user wants to authenticate to use web services features offered on a web site, the web site redirects him to a SWC authentication page. The authentication page uses the id of the client to warn the user that the specified client is trying to authenticate him and displays a login form (Step 1a). Once successfully authenticated,

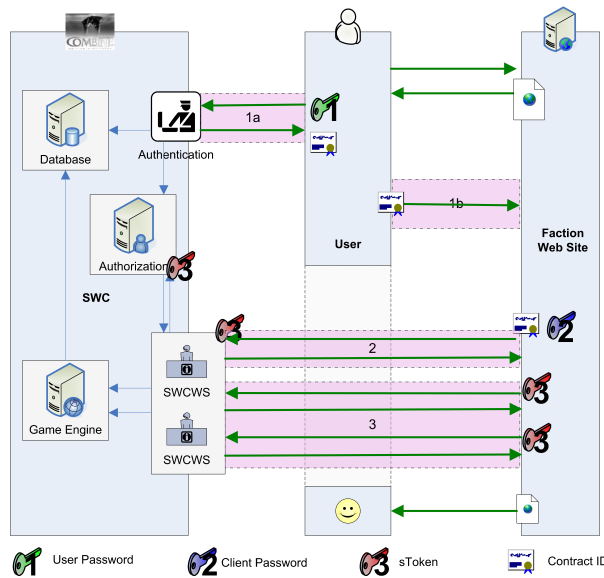


Figure 4: Authentication process overview

the SWC server generates a contract. The contract includes information such as the user identification number, the client identification number and the contract validity duration. Then, the server redirects the user to the web site. It uses the user's browser to communicate to the client a contract identification string. If the client is not registered as a web interface, the authentication page simply displays the identification string (Step 1b).

2. **Client authentication and generation of a security token:** In order to work, SWC web services require security tokens, briefly sToken. The sToken must be provided for each web services call by the client. A sToken is an object abstraction of a few security information such as the contract identification string or the query number. In order to obtain a valid sToken, the client has to request one from the SWC server using a web service (Step 2).
3. **Accessing web services using a sToken:** Using the sToken received in Step 2, the client can try to access web services. The client simply has to provide the sToken in every web service call it makes. All communications are encrypted using SSL, this prevents third parties to access the exchanged informations (Step 3).

5.2 Authorization

The SWC web services are grouped in the following three categories:

- **Public web services:** These web services do not require any kind of authentication. Their purpose is

advertisement. They are really easy to implement but do not offer many functionalities. They can be used to obtain server time, time conversion and the list of all visible systems in the galaxy or estimate time of arrival for a given travel.

- **Client-restricted web services:** These web services require only client authentication. They have more functionalities than public web services: clients can download the whole galactic map, browse the character profiles or connect to the galactic news system.
- **User-restricted web services:** These web services require both client and user authentication. Players can download their character information, list the faction members, connect to their personal inventory or retrieve their faction assets.

The authorization process has two main objectives: allowing administrators to introduce restrictions aimed at preventing abuses (via so-called administrator rules) as well as allowing users to introduce restrictions aimed at protecting their privacy (via so-called user rules).

The administrators can introduce restrictions via rules of the following types:

- **User rules:** Such rules have the finest granularity. They allow the administrator to authorize or deny web service access to a particular character using a specified client. A user-type rule requires that user authentication is used in order to apply.
- **Factions rules:** Such rules have a lower granularity as they manage all the members of a faction at the same time. They are only effective if no user rule was found and if user authentication is used.
- **Clients rules:** Client rules have a lower priority and granularity than the two previous ones. They manage all connections made by anyone trying to access particular web services with a given client.
- **Default rules:** Default rules have the lowest priority and granularity of all. They allow the administrator to set up a default behavior when accessing a web service.

Additional rules have to be set up by the user. These rules ensure that the user has deliberately allowed a client to access a web service using its account.

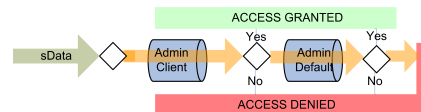


Figure 5: Authorization rules for WS requiring client authentication

Figure 5 represents the different steps the authorization goes through before access is granted or de-

nied to a particular client. In case of user authentication, the user needs to authorize the client to use the web services. The authorization process uses four filters to get to a decision as illustrated in Figure 6.

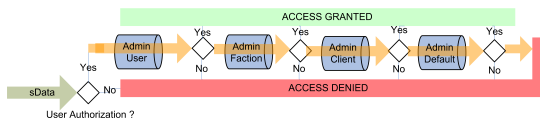


Figure 6: Authorization rules for WS requiring user authentication

Although this system allows a very fine control of what can be done or not, it got some negative feedback. The user authorization steps were seen as too complicated, as many users were not aware of which web services are involved in most transactions. As an improvement, users proposed the creation of a package concept. Packages are sets of web services that are working in the same semantic domain. For example, an inventory package was created to group all the web services related to the various assets owned by a faction. For each package, a description is available. Furthermore, for each web service in a package, a user friendly description is provided.

6 COMMUNITY ACCEPTANCE

One major objective of this project is to observe how the SWC community responds to the creation of the web services infrastructure. Measuring community acceptance of new features, such as the web services proposed in this paper, in a massively multi-player on-line role-player game is a difficult task. The measurement includes many elements, which need to be analyzed by various specialists such as psychologists, sociologists, game experts, etc. However, the following indicators predict a good future for a wider adoption of web services in the MMORPGs context:

- The large number of clients developed or being developed over a very short period of time.
- The numerous complaints about the imperfections in the pilot web services architecture proposed.
- The huge number of requests coming from the developers for new web services features.
- The high number of utilizations revealed via the statistics tool.
- The initiative taken by some developers to create a wiki web site dedicated to document the web services component and share their code.
- The great speed at which bugs have been identified.

So far, twenty two clients went through the registration process successfully. The Galactic Empire is

one of the largest faction in the game and provides all its members with a single and user-friendly web page containing all the galactic news.

The CenterPoint Space Station, another large faction, is using web services to keep its inventories up to date. Once a player has performed authentication through the web services mechanism, he has access to all his character's belongings and he can also obtain a list of his faction's assets, provided he has the required privileges. Furthermore, the CenterPoint Space Station has created a dynamic map to locate its assets in the galaxy. Assets are marked with dots on the map. When moving the mouse to one of these dots, the user can obtain various information about the assets placed there.

One of the main activities of the Galactic Corporation consists of an on-line auction web site for objects in the galaxy. It is concerned with many security issues, in particular the verification of its user accounts. Not surprisingly, the Galactic Corporation has mainly been using account verification web services.

Every time a client is successfully authenticated and authorized to use a web service, a record containing its clientID, the userID, the time and the web services requested is stored on the database. All these data are then used for statistics purposes. Figure 7 shows the web services use from 16 June to 8 August 2005. The huge amount of calls made by the Galactic Empire in order to access the news system should be noticed. Security measures have to be taken to avoid massive resources consumption by a minority of clients. An automated limitation tool will need to be put in place so as to avoid abuses or unfortunate incidents.

7 CONCLUSIONS

Today, SWC web services have reached the stage of direct trust relationship defined in Section 4. The large number of clients developed in a very short time has demonstrated the usability of the technology. The current release of the SWC web services component allows clients to retrieve information about inventories, characters, galaxy map, galactic news and time as structured data easily integrable in any other applications.

While building the web services server application and proving its usability and usefulness was quite an achievement, the community acceptance objective was not forgotten. We think this objective was also reached, as demonstrated by the comments received from users.

As for the future of the web services component, we strongly believe that it can easily be expanded to include new features. As the infrastructure is now in

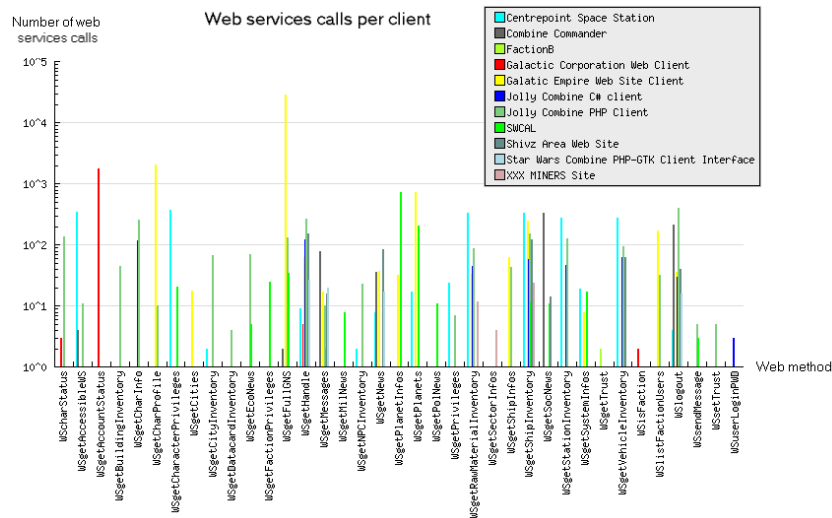


Figure 7: SWC web services usage

place, this would not take up much time. On the other hand, it would be very useful to set up a developer working group to insure that the new features are in line with the game rules.

Concerning the development of the next version that would allow to reach the indirect trust relationship stage, we, however, recommend to pay great attention to the following aspects:

- the important amount of time which will be needed to set up the necessary public key infrastructure;
- the client technical limitations, as most clients will not be allowed to install the cryptographic libraries required; and
- the high level skills that will be requested from the developers, in particular on the client side.

For these reasons, we do not recommend that the current web services component be upgraded in the short time. SWC is already a pioneer with regard to web services: no other game is yet using such a technology. We think it would be wise to wait for more feedback from the community before investing more time in web services.

ACKNOWLEDGMENTS

We would like to thank Jehan Snyers d'Attenhoven for imagining the SWC and without whom this passionating work wouldn't have been possible, and also all the SWC developers for their devoted work and patience.

REFERENCES

- Daconta, M. C., Obrst, L. J., and Smith, K. T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley.
- Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Kroghdahl, P., Luo, M., and Newling, T. (2004). *Patterns: Service-Oriented Architecture and Web Services*. IBM Redbooks.
- Fuller, J., Egervari, K., Fuecks, H., Waters, B., Stephens, J., and Solin, D. (2003). *Professional PHP Web Services*. Wrox.
- Graham, S., Davis, D., Simeonov, S., Daniels, G., Brittenham, P., Nakamura, Y., Fremantle, P., dieter Konig, and Zentner, C. (2005). *Building Web Services with Java, Making sense of XML, SOAP WSDL, and UDDI*. Sams Publishing.
- IGDA Online Games SIG (2004). *Web and Downloadable Games White Paper*. http://www.igda.org/online/IGDA_WebDL_Whitepaper_2004.pdf.
- Iverson, W. and O'Reilly, T. (2004). *Web Services in Action: Integrating with the eBay Marketplace*. O'Reilly.
- OASIS (2005). *Security Services (SAML) TC*. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.
- Rosenberg, J. and Remy, D. (2004). *Securing Web Services with WS-Security: Demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Sams Publishing.
- Sarang, P., Browne, C., Ayala, D., and Chopra, V. (2003). *Professional Open Source Web Services*. Wrox.
- Snyers d'Attenhoven, J. (2003). *SW Combine Strategic Overview*. <http://www.swcombine.com/documentation/pdf/SWC-SO-v1.0.pdf>.