

# A Scalable Distributed Peer-to-Peer MMOG Architecture

John S. Gilmore  
 Department of Electronic Engineering  
 Stellenbosch University  
 Stellenbosch, South Africa  
 Email: jgilmore@ieee.org

## Abstract

The abstract goes here.

May, 2010

## I. INTRODUCTION

WITH the advent of broadband Internet, Massively Multiplayer Online Games (MMOGs) have gained significant popularity over the course of the past few years. Figure 1 shows the total number of active MMOG subscriptions over time for the period 1997 to 2008. From here, the accelerating growth of the MMOG market is evident.

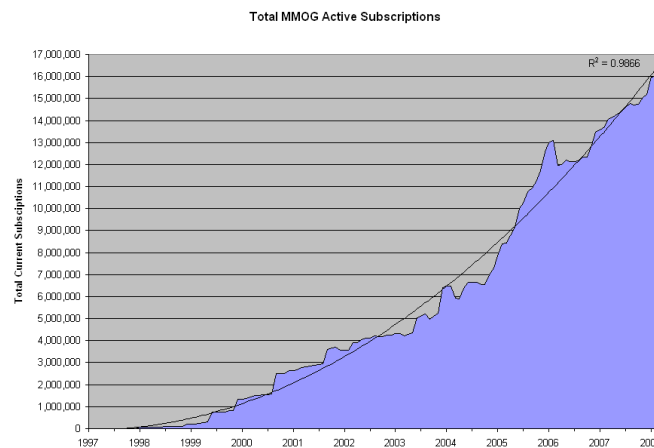


Fig. 1. Total number of active MMOG player subscriptions over time [1].

MMOGs are characterised by expansive worlds, where a large number of players interact online with each other and the virtual environment to achieve certain goals through collaboration and teamwork. Throughout the development of MMOGs, role play has been tightly coupled to this type of game. This is perhaps due to the exploration and player interaction aspects. Role play allows players to fully immerse themselves in the game world and might, therefore, provide for a more compelling experience. Because of this tight coupling, the terms Massively Multiplayer Online Role Playing Game (MMORPG) and MMOG have almost become synonymous. Throughout this work, a distinction will, however, be made between the two.

MMOGs hold great commercial as well as academic value. From a commercial perspective, the growing number of active subscriptions shown in Figure 1 translates to a growing MMOG market. Figure 2 shows the online games market forecast by DFC Intelligence, a company specialising in game market forecasts for various sectors.

From an academic perspective, MMOGs also hold great value. An MMOG is a complex networked application, with clients requiring reliable real-time feedback on actions taken. The design of an MMOG requires in-depth knowledge of server architectures and network design. The design of a server architecture determines how many players the game will be able to host and what the user experience will be in terms of quality of service. The server architecture must be able to handle thousands of requests, store large amounts of data, update the game state and send responses back to all clients in real time.

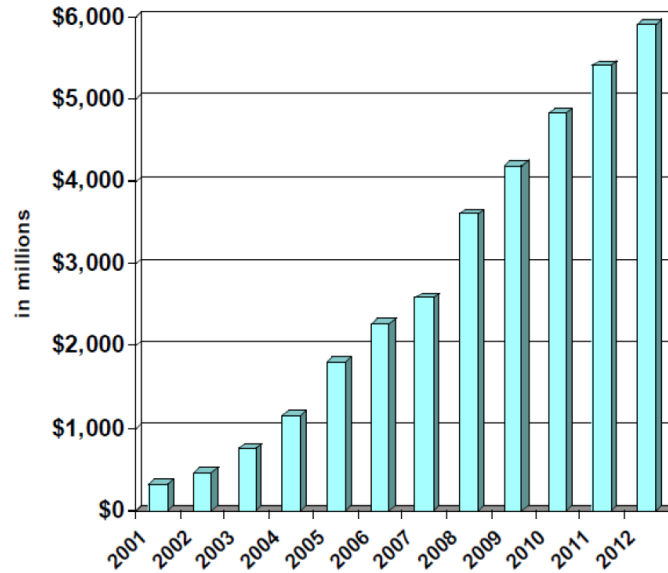


Fig. 2. DFC Intelligence MMOG market forecast '08 [2].

## II. A BRIEF HISTORY OF MMORPGS

MMOGs have a history stretching from 1978 to the present. A complete history of MMOGs and the histories of the games and boardgames that they originate from can be found in [3] and Chapter 1 of [4]. The first games that could be called MMOGs were Multi-User Dungeons (MUDs) (1978) [5]. MUDs are entirely text-based, with players exploring areas by receiving descriptions of what they were “seeing” and typing commands to move and interact with objects or players. MUDs contain many of the elements that today are still central to the MMOG concept. These elements include exploration, large worlds, multiplayer, social interaction and progression.

After MUDs, there were many MMORPGs that acted as building blocks for what is recognised today as being an MMORPG. The first graphical MMORPG was *Neverwinter Nights* (1991) [6]. *Neverwinter Nights* was not an Internet based game, it was hosted on what today is the AOL network. *Meridian 59* (1994) was the first MMOG to have featured a monthly subscription fee, receive wide media coverage and most importantly, the first MMOG to feature a 3D engine [7].

The first MMORPG to be commercially successful and largely credited with popularising the genre was *Ultima Online* (1997). *Ultima Online* used existing intellectual property from the *Ultima* universe as well as an aggressive marketing campaign by game publisher EA, to quickly gain 100,000 subscribers. NCsoft’s *Lineage* (1998) looked similar to *Ultima Online*, but was more Player-versus-Player (PvP) oriented and with an added castle siege mechanism, became very popular in South Korea. *Lineage* had more than three million subscribers at one point, most of them from South Korea [8]. *Everquest* (1999) is credited for bringing MMORPGs into mainstream Western Culture. It featured a large persistent 3D environment that was capable of hosting up to 15000 players per server [9].

MMORPGs in the first millennium are considered to be of the first generation. These games provided blueprints for all MMORPGs to follow in the second generation. While there are many more games in the second generation, these games are characterised by little innovation in the genre and focus more on improving graphics and ease-of-use [3]. Notable games in this generation are: *Final Fantasy XI* (2000) (console based), *Dark Age of Camelot* (2001) (realm vs. realm combat) and *Anarchy Online* (2001) (instancing).

*Eve Online* (2003), developed by CCP Games in Iceland, brought many new innovations to the MMORPG. It was the first successful MMORPG to feature a science fiction theme. It was the first MMOG to have a single distributed server architecture. This meant that no sharding was required. Sharding is a method by which the game world is duplicated onto multiple servers to distribute load. Players cannot communicate between shards as these worlds are completely isolated from each other. By employing a distributed server architecture, where different regions of the virtual world was hosted on different servers, players could have a seamless and more immersive experience. This was accomplished by hosting different star systems on different servers. Players have to use a warp gate to travel between star systems. This mechanic is used to mask the time it takes to move the player from one server to another. In 2006, CCP Games launched the largest supercomputer in the gaming industry to upgrade their existing infrastructure and enable *Eve* to support more than 50,000 concurrent users [10]. This number was surpassed in 2009 with 54,181 concurrent users in game [11].

Another innovation of *Eve* was the in-game economy. CCP games appointed Dr. Eyjólfur Guðmundsson as chief economist of *Eve online* in 2006 [12]. His duties were to monitor and predict market trends in the game world and produce detailed quarterly economic reports [13]. The economy is based on a open market system ruled by supply and demand. No other game

has implemented an in-game economy in such a rigorous fashion.

Blizzards's World of Warcraft (WoW) (2004) is the most successful MMORPG to date. After six years it still has by far the most subscribers of any MMOG, totaling 11,5 million, each paying \$15 per month subscription [14]. In 2008 it was estimated that WoW holds more than 60% of the MMORPG subscription market [15]. From the first generation of games, a steady growth has been seen in the MMOG space, but before the runaway success of WoW, no one had estimated that the gaming market could be this large [16]. It should be noted, however, that the growth seen in the MMOG market, is mostly due to growth in the Asian markets and that the size of these markets are much larger than the size of the Western markets.

The success of WoW has largely been attributed to the overall quality and finish of the game [17]. It is interesting to note that WoW is not attributed with many innovations. Most games that came before it implemented most of the features in WoW. What WoW did do, is combine all previous innovations into a package that was accessible to a large number of people. Players also don't just play WoW to experience the game content, they also play the game to meet up with friends and socialise. Guilds are also an integral part of WoW. Guilds are collections of players that choose to play together to achieve some common goal.

### III. OVERVIEW OF MMOG NETWORK ARCHITECTURES

The previous section presented a brief history of the MMOG and more specifically, the MMORPG genre. This section explores the network architectures that are generally employed by this genre of games. The network architecture of a system specifies how the different entities in the network communicate, where authority lies and whether a centralised or distributed approach is followed. Currently, all MMOGs being operated run on a Client/Server (C/S) architecture or a distributed C/S architecture, also called a Client/Multi-Server (CMS) architecture. The simplest form of network architectures used in MMOGs is the pure C/S model.

Figure ?? shows the C/S model. The server is the entity on which the MMOG is hosted and is controlled by the game producer. Clients are computers operated by players, that connect to the server to play the game. The server is responsible for updating the game state, according to client actions, to store the new game state and to distribute game state updates to all clients. An in-depth exploration of consistency models, of which this theory forms part, will be presented in Section VI. Clients never communicate with other clients. They send their actions to the server and receive the updated states of other players from the server.

The C/S architecture has two main advantages that made it the architecture of choice for all MMOG developers. Because of the centralised approach of the architecture, both administration and security are greatly simplified. Administration is simplified, because the game producer has full control over the server, server data and code. It controls physical access as well as which clients are allowed to connect to the system. Efficient logging is also supported, because the server is able to not only log all server actions, but also all client actions, since these also pass to the server.

Security is a significant issue in MMOGs, since some players sell in-game currency for real-world currency []. This makes the MMOG a platform that is capable of producing income, which increases the incentive of players to be able to gain an unfair advantage over other players. The more popular an MMOG, the greater the security threat. Because the producer has full control over the server code and is never required to furnish the client with the server code, it makes it difficult for a potential attacker to use knowledge of the server in an attack. Because clients are never allowed to communicate, all malicious users can be filtered out of the network by the server when detected and even banned from the network. Producers are able to ban players, because these games usually require a game account, which is linked to a copy of the game as well as some payment method. This introduces a large cost to player whose accounts are banned.

The C/S architecture however does have some disadvantages. These are weak robustness, weak scalability, high cost to the provider, high latency, high amount of required server bandwidth and weak handling of transient loads. The robustness of the system is weak because it is a single point of failure. If the server fails or goes down for maintenance, the game is off-line and players are unable to play.

This system is also not scalable, since a single server cannot easily be extended with more resources. Even if an off-line approach is used, where hardware is upgraded after the system is taken down for maintenance, the hardware required to support a game hosting more than 300 players, become prohibitively expensive []. The problem leads to the weak handling of transient peak loads. Players prefer playing during certain times and usually after a game patch has been released, the players load increases for a short period of time. The server hardware should be able to support peak system loads, which means that sufficient resources should always be provisioned to support these peak load. This is not an economically viable solution, because resources to handle peak loads are not used most of the time. This translates to producers paying for the provisioning of resources, without having active players that pay for these resources.

This disadvantage also leads to the next one, namely high cost for the provider, which translates to high costs for clients or reduced profit. The cost come from the hardware that is required to host the system as well as maintenance and running costs. Running costs include bandwidth provisioning and IT personnel and maintenance costs include replacement of malfunctioning or outdated hardware. It has been estimated that maintenance and running costs consume approximately 80% of game revenue during the lifetime of the game [].

Because no clients are allowed to communicate with other clients, every change that is made to the game world by a client, first had to be communicated to the server, which in turns relays this message to all clients after applying game logic and

artificial intelligence (AI) algorithms. This two hop path, with the additional time for computation added by the server as well as possible buffering at the server when many clients communicate, significantly increases the latency of the system compared to a system where direct communication is used.

In an effort to address some of these issues, the distributed C/S, also called C/MS, was introduced. In a distributed C/S model, the server functions are distributed amongst multiple machines in an effort to distribute the server load. For MMOGs, there are various methods employed to achieve this. One of the first methods used is called sharding, where copies of the game world run on different servers and clients connect to one of these shards []. Clients are not able to interact or communicate with players on other shards, which reduces game immersion. This method does, however, allow for a more scalable system as maximum load is fixed. Players are not able to enter a shard if that shard has reached its capacity. This has in the past caused unhappiness amongst players, since popular shards could be difficult to log in to. Players are also reluctant to move to a new shard, because a lot of time is invested in their characters in their “home” shard. For all practical purposes, this approach is still merely a C/S approach, with players forced into specific C/S environment.

There are three popular true distributed models used in MMOGs today. These are redundancy based, region based and object based []. The first one is very similar to sharding, with the difference that all servers share the same duplicated game state. Each server contains the global game state and clients connect randomly or through some sort of load distribution algorithm to a server. Each server handles all actions from clients and updates its own database. The servers in turn send updates to each other over a high quality link, such as fibre, to maintain database consistency at high speeds. The problem with this system is that the world is never truly consistent and that there are no optimally chosen inconsistency obfuscation boundaries. In other words, two players standing next to each other in the virtual world, might be on different servers and, therefore, experience two different worlds. Game inconsistency is not necessarily an issue, but then it should be based on a distance based approach where the consistency degrades gracefully the greater the distance between players in the virtual world. This system is also not truly scalable because of the large hardware costs involved in the system as well as network hardware required to achieve sufficient performance for large numbers of players.

Another method that has also been employed is to divide the virtual world up into regions and host the different regions on different servers []. Busy regions are hosted on their own servers, while multiple quiet regions are hosted on a single server. The issue of this static region approach is that it also does not scale well when one region is suddenly populated with players. This type of behaviour happens quite regularly and is known as flocking []. When players find something of interest in a region, many players will flock to that region. In-game events and festivals are also becoming popular and these events also cause flocking to the region where the event is held. The solution to these effects have been over provisioning of resources to handle peak loads, which suffers from the disadvantages discussed above. Also, if the load changes, the server has to be brought off-line in order to rebalance the regions. Dynamic regions are being investigated, where regions can be dynamically shifted from one server to another, in order to balance load []. This approach unfortunately causes a lot of overhead and adds significant complexity with regards to the migration of the data and the handling of player actions while the data are in transit [].

The third method used is to distribute all in-game objects amongst the servers. For an MMOG, most of these objects are expected to be players objects. The advantage of this method is that the system load is fixed for a certain player population and that the load is equally distributed amongst all servers. This allows for more accurate prediction and provisioning of required resources, but still does not handle transient loads well. Another issue is inter-server communications for this architecture. The inter-server communications are random and also much more than the inter-server communications for a region based system. The reason for this is that the amount of player interaction increases with a decrease in the distance between the players. Players playing together move together, chat and interact with Non-Player Characters (NPCs) together. For a region based model, all player interaction near to each other remain in the server.

In general, the issues addressed and improved by the C/MS architecture are robustness, scalability, and peak load handling. The system is more robust, because the failure of one server will not necessarily lead to the failure of the whole system for certain system designs. The system is more scalable, because many less powerful servers may be used, which allows for the hosting of more players than what is currently possible with single server hardware. It also handles transient loads better, because, for cases where loads can be predicted, resources can be shifted between servers to improve the user experience.

The disadvantages of this system is that the administration complexity is greatly increased. Such systems, although capable of handling many more users than a single server, is also much more expensive. These disadvantages are however not technical problems and so it is assumed for current games, that these systems are what is required if a game is to be hosted for a large number of players.

## IV. PEER-TO-PEER MMOG ARCHITECTURES

### A. Overview

Recently, an alternative architecture has gained popularity, making use of a peer-to-peer architecture to host the MMOG []. This has opened up a new research field that is attempting to make the peer-to-peer model a viable alternative to the classic C/S and C/MS architectures. This architecture does, however, still have a few major issues that need to be solved before MMOGs

can be developed that use it. If these issues, which are later discussed, can be solved, a Peer-to-Peer (P2P) architecture holds some powerful advantages over a C/S system.

The core idea of the P2P model is that each peer contributes enough resources to the network to be able to host itself. This also means that all functions of the server in the classic C/S model are distributed amongst all peers. There are many areas where the P2P model can improve on the classic C/S model. These areas are robustness, scalability, provider cost, latency, server bandwidth and handling of peak transient load.

The system is very robust, because there is no server that can fail, only individual peers. Individual peers failing will not affect any other peers other than the peer that failed. This behaviour makes game down-time extremely unlikely.

Also, because every peer is expected to add sufficient resources to the system to host itself, this makes the system very scalable with no extra costs being incurred from a provider viewpoint for any peer that joins the network. This will also allow for efficient handling of transient loads. If many players suddenly enter the game, no resourcing provisioning issues will arise, as peer bring the resources they need with them into the system. It should also be noted that it is not at all unreasonable that a peer will have sufficient resources to host itself. Peer computers are very powerful systems these days, with multi-core CPUs, multiple gigahertz of clock speeds, multiple gigabytes of memory and secondary storage space in the terabyte range. The graphics cards in gaming machines have also become immensely powerful. With such a system in mind as the average gaming system, it is not unreasonable to assume that sufficient resources will be available.

P2P architectures also create a lot of opportunity for independent developers, because a large initial investment is now no longer required to purchase the expensive server hardware. Not just are hardware costs greatly reduced, but running costs are also greatly reduced. The bandwidth required by the game server is now shared amongst users. Which means that no bandwidth costs will be incurred by the provider. Also, the amount of required bandwidth per user is usually not high, which means that users will not be expected to use a much greater amount of bandwidth.

Latency is also improved, because it is now possible to directly communicate between peers and not necessary to go through a server for communications. There is also no single server that has to process peer actions. Peer actions need only be processed by other peers who find the specific peer actions of interest. The distribution of the load as well as direct communication will reduce latency.

Property	Client/Server	Client/Multi-Server	Peer-to-Peer
Administration	Simple	Challenging	Complex
Security	High	High	Low
Robustness	Low	Medium	High
Scalability	Low	Medium	High
Provider cost	High	Very high	Low
Latency	High	High	Low
Server bandwidth	High	High	None
Peer bandwidth	Low	Low	Medium
Peak transient load handling	Bad	Medium	Good

TABLE I  
DIFFERENCES BETWEEN CLIENT/SERVER, CLIENT/MULTI-SERVER AND PEER-TO-PEER ARCHITECTURES

Table I summarises the differences between the three architectures as discussed thus far. What remains to be said is why administration and security were marked as complex and low respectively. From an administrative perspective, it is more complex to administer a decentralised system than a centralised one []. The reason is that the game producer has full control over the server in the C/S architecture. The administration tools only have to control the centralised server or server cluster. For a peer-to-peer system, the server is the collection of all peers. The game producer does not have direct access to peers and so controlling these nodes become significantly more complex.

The server or cluster is housed in a secured location, where access can be controlled. As stated earlier, the server code is also protected and not accessible to potentially malicious players. These factors simplify the security of the C/S model by allowing the developers to place all intelligence in the server and move all sensitive computations to the server. Clients do not house game logic and the client state is also not authoritative.

If this model is contrasted with a P2P model, the P2P model has some significant security challenges []. The challenges reside in the fact that peers are not under the control of the game producer. Since all server data are distributed amongst peers, all peers have access to sections of the server data. Peers also have access to the distributed server code. One advantage that can be exploited is that no peer contains all server data and no one peer has more authority than another. This allows for schemes such as quorum algorithms to be used to increase the security level. This and other techniques are discussed in Section ??.

### B. Key challenges and requirements

In past literature, key requirements and challenges have been identified that a successful P2P architecture will have to address. The six key challenges identified of such a system are: interest management, event dissemination, security, state persistency,

NPC hosting and authentication []. Keep in mind that these challenges are not unique to a P2P system, but their solutions are. The key requirements are [].

Interest management is used to determine the smallest amount of information that a peer requires, in order to present an accurate representation of the world to each player []. The main idea is that a player has a limited visual range and a limited area around the player in which it can interact with objects. The player requires update information of all objects in this area, called the player's Area of Interest (AoI). AoI calculations also rely on the fact the a player's direction and velocity of movement also cannot change instantaneously and are bounded. Extensive research has been done into solving AoI problems. The solutions range from aura/nimbus [] to publish/subscribe [] to Voronoi diagram models []. Generally, interest management solutions can be divided into coarsely or finely grained solutions.

Coarsely grained solutions usually divide the game world into multiple regions and when a player enters a region, it subscribes to that regions events. All players in the region then receive the region's events, even for players not in their AoI. Finely grained techniques create groups of players from their AoIs. Groups of interacting players directly exchange information, so all players only receive information that is relevant to them. The grain of the solution in turn determines the type of event dissemination that should be used [].

Event dissemination deals with how information should be sent to peers after interest management has determined which information should be sent []. Recently, Application Level Multicast (ALM) and unicast techniques of event dissemination have become popular, depending on the grain of the event dissemination. ALM is used, instead of router level multicast, because of a lack of general support for this technology at the router level [].

ALM is used for coarsely grained interest management techniques, while unicast is used for finely grained techniques. Unicast is not used for coarsely grained techniques, because it's not scalable [], but ALM significantly increases the message latency in the system, because messages first have to be routed over a P2P overlay network []. ALM is however preferred over unicast for large numbers of messages, because of the weak scalability.

Security has been identified as a major issue for P2P systems []. This is because of the lack of control of peers. There are various security issues and these are usually divided according to the level of the protocol stack where they occur. The areas that have been identified by [] and expanded upon by [] are: game level, application level, network level and infrastructure level. Game level threats are usually referred to as cheats. These are ways in which a malicious players may gain an unfair advantage over other players, within the confines of the game. These cheats are usually because of software bugs and some examples are duplication and teleport cheats.

Application level threats are where malicious players alter the game software to gain an unfair advantage. This is usually done by gaining access to the game state to which they should not have access at the current time. An example of this is map reveal cheats in strategy game. Where the fog of war is removed and the player can observe all the opponent's movements. Other cheats are sometimes used that augment the player's User Interface (UI) with extra information that allows the player to make more informed decisions. It is debatable whether these additions are cheats, but they are considered almost essential for competitive WoW play [].

For P2P threats, securing the network level is of great importance and mainly focusses on securing the IP level network. The field has recently received more attention than in the past and has started to bear fruit []. The solutions concern securing the system against classic IP layer threats such as delaying packets in transit, dropping packets and altering packets. Handling the alteration of packets is a specialised field in security and effectively attempts to make the system Byzantine fault tolerant [].

In the P2P overlay, replication with quorum (voting) [] and erasure codes [] are used in order to secure the system against malicious nodes, altering objects. This is, however, an ongoing research field with many issues still open. Open issues include ensuring reliable system performance under collusion attacks.

## V. PROPOSED ARCHITECTURE

## VI. CONSISTENCY MODELS

## VII. PROPOSED STATE PERSISTENCY MODEL

## VIII. RELATED WORK

## IX. CONTRIBUTIONS

## X. CONCLUSION

The conclusion goes here.

## REFERENCES

- [1] B. S. Woodcock. (2008) Total MMOG active subscriptions. mmogchart.com. [Online]. Available: <http://www.mmogchart.com/Chart4.html>
- [2] L. Fan, "Solving key design issues for massively multiplayer online games on peer-to-peer architectures," Ph.D. dissertation, School of Mathematical and Computer Sciences – Heriot-Watt University, May 2009.
- [3] L. Achterbosch, R. Pierce, and G. Simmons, "Massively multiplayer online role-playing games: the past, present, and future," *Comput. Entertain.*, vol. 5, no. 4, pp. 1–33, 2007.



- [4] R. Bartle, *Designing Virtual Worlds*. New Riders Games, July 2003.
- [5] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit MUDs," *Journal of MUD Research*, vol. 1, pp. 1–25, 1996. [Online]. Available: <http://www.mud.co.uk/richard/hcds.htm>
- [6] Medar. (2004, October) The history of Neverwinter Nights. [Online]. Available: <http://www.bladekeep.com/nwn/index2.htm>
- [7] A. Kirmse. (2000, May) History of Meridian 59, 1994-2000. [Online]. Available: <http://sites.google.com/site/meridian59history/>
- [8] B. S. Woodcock. (2008) MMOG active subscriptions 200,000+. mmogchart.com. [Online]. Available: <http://www.mmogchart.com/Chart1.html>
- [9] T. Fritsch, H. Ritter, and J. Schiller, "The effect of latency and network limitations on MMORPGs: a field study of Everquest 2," in *NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*. New York, NY, USA: ACM, 2005, pp. 1–9.
- [10] M. Bergsson and J. Alberni. (2006, September) Eve online launches largest supercomputer in the gaming industry running on IBM server technology. CCP Games and IBM. [Online]. Available: <http://www.eveonline.com/pressreleases/default.asp?pressReleaseID=25>
- [11] C. Manifest. (2009, December) New PCU record of 54,181 pilots. CCP Games. [Online]. Available: <http://www.eveonline.com/news.asp?a=single&nid=3620&tid=1>
- [12] V. Massey. (2007, June) Eve online appoints in-world economist. CCP Games. [Online]. Available: <http://www.eveonline.com/pressreleases/default.asp?pressReleaseID=34>
- [13] E. Guðmundsson (edt.). (2010, February) Eve Online fourth quarter economic newsletter for 2009. CCP Games. [Online]. Available: [http://ccp.vo.llnwd.net/o2/community/QEN/QEN\\_Q4-2009.pdf](http://ccp.vo.llnwd.net/o2/community/QEN/QEN_Q4-2009.pdf)
- [14] B. Entertainment. (2008, December) World of Warcraft subscriber base reaches 11.5 million worldwide. Blizzard Entertainment. [Online]. Available: <http://us.blizzard.com/en-us/company/press/pressreleases.html?081121>
- [15] B. S. Woodcock. (2008, April) MMOG subscriptions market share. mmogchart.com. [Online]. Available: <http://www.mmogchart.com/Chart7.html>
- [16] B. S. Woodcock. (2008) An analysis of MMOG subscription growth version 23.0. mmogchart.com. [Online]. Available: <http://www.mmogchart.com/analysis-and-conclusions/>
- [17] N. Ducheneaut, N. Yee, E. Nickell, and R. J. Moore, "Building an mmo with mass appeal: A look at gameplay in world of warcraft," *Games and Culture*, vol. 1, pp. 281–317, 2006.