

Technologies and Mechanics behind Virtual  
Worlds  
final report

Tim Waage  
Center for Informatics, University of Goettingen  
Seminar On Internet Technologies, SS08  
Email: [tim.waage@web.de](mailto:tim.waage@web.de)

September 30, 2008

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Characteristics of Virtual Worlds</b>	<b>4</b>
<b>3</b>	<b>Virtual Worlds based on client-server-networks</b>	<b>5</b>
3.1	Advantages . . . . .	5
3.2	Disadvantages . . . . .	5
3.3	Sharding . . . . .	6
3.3.1	How it works . . . . .	6
3.3.2	Example: "World of Warcraft" . . . . .	6
3.3.3	How to avoid it . . . . .	7
<b>4</b>	<b>Virtual Worlds based on peer-to-peer-networks</b>	<b>8</b>
4.1	Advantages . . . . .	8
4.2	Disadvantages . . . . .	8
4.3	Special P2P security considerations . . . . .	8
4.3.1	models of trust . . . . .	9
4.3.1.1	Hierarchic PKI . . . . .	9
4.3.1.2	Cross certification . . . . .	10
4.3.1.3	"Web of Trust" . . . . .	10
4.3.2	New approaches to extend PKI . . . . .	10
4.3.2.1	SPKI . . . . .	10
4.3.2.2	SDSI . . . . .	13
4.3.3	Threats for P2P-networks . . . . .	14
4.3.3.1	Man-in-the-middle attacks . . . . .	14
4.3.3.2	Incorrect Routing Updates . . . . .	14
4.3.3.3	Partitioning . . . . .	15
4.3.4	Possible counteractive measures . . . . .	15
4.4	Frameworks for virtual worlds in P2P-networks . . . . .	16
4.4.1	NICTA . . . . .	16
4.4.2	Vast . . . . .	16
4.4.3	Solipsis . . . . .	17
<b>5</b>	<b>Conclusion and personal thoughts</b>	<b>17</b>
<b>6</b>	<b>References</b>	<b>18</b>

## Abstract

*Today Virtual Worlds are mainly 3D-Online-Infrastructures. The most important goal is the creation and simulation of a so called "Metaverse", a word that first appears in Neil Stephenson's Sciencefiction Novel "Snow Crash" [1]. It describes a world similar to the reality, where people are able to interact with each other represented by avatars. They can also move around or use and combine virtual objects. Lots of things are needed to be simulated in order to fill a Metaverse with virtual life, for example physics or artificial intelligence of non-player-characters. By now there are many 3D virtual environments based on this concept. A few of them will be discussed in this report, each one representing another approach to deal with the special requirements coming along with a virtual environment.*

*Every virtual world has a different scenario and some techniques are more suitable for certain aspects of it than others, but there are some things that all sorts of virtual worlds are having in common. First of all they are based on either client-server-networks or peer-to-peer-network. Secondly they have to deal with a huge amount of participants, which is why they are producing a lot of data that has to be processed somehow in nearly realtime. And last but not least there is very much almost constant data (for instance the worlds terrain data or user profiles) that needs to be stored somewhere.*

*To make all that work and especially to guarantee a high level of speed, security and trustworthiness, every network representing a virtual world has to be equiped with the according features in order to keep the virtual world attractive to a large number of users (and potential customers). Hence there are more and more P2P-based Virtual Worlds lately this report will focus on them especially.*

*Keywords: client-server-network, peer-to-peer-network, Virtual World, sharding, SPKI, SDSI, World of Warcraft, Second Life*

## 1 Introduction

The variety of Virtual Worlds is immense. They appear textual (forums, blogs, wikis...), two-dimensional or with today's computing power mostly three-dimensional. The market is dominated by Virtual Worlds, in which hundreds of thousands of people can participate. They create millions of transactions every day by interacting with each other (for instance chatting, fighting...) or with the computer-simulated environment (for instance acting with objects or non-player-characters).

Different Virtual Worlds have different things they focus on. Today's most important classes of Virtual Worlds are social networks and so called MMORPGS (Massively multiplayer online role-playing games) with a incredible huge amount of players. Both "World of Warcraft" [2] as most famous MMORPG and "Second Life" [3] as most common social network have more than 12 million active subscriptions each [4].

## 2 Characteristics of Virtual Worlds

Similar to the real world a virtual world is also supposed to be accessible 24 hours a day, seven days a week. Compared to other applications virtual worlds require long connections times with permanent data streams, because it is quite normal to have very long sessions. In other types of multi-user applications the duration of a single session is much shorter. For instance a session in ego-shooter games often takes just a few minutes.

The requirements for delays are also different from the most other applications. For a participant in a virtual world it is quite okay to have a ping of about 300-500ms. It is still playable without any big negative impacts on the quality the participants experience the virtual world. Compared to the ego-shooter example from the first paragraph, where the ping should not cross the 50ms threshold, this is quite a lot. But more than one second delay is not tolerable. The experience would suffer.

It is also undoubted, that virtual worlds can create very huge amounts of data, that needs to be proccesed and stored in some way. Users have their own personal inventory, that means there is a need for a database containing items or other self created objects (for instance "Second Life" uses MySQL databases [5]). Changes in the terrain or landscape of the world are also needed to be saved, stored and made visible to other users.

Security considerations are becoming more and more important as well. Some virtual worlds offer many possibilities to spend real money or do other things that can affect the real world. For instance, in "Second Life" there are many shops representing enterprises well known in the real world, where users can purchase items for their virtual charakter or for themselves. Therefore it is necessary to transmit data entirely secure und protect it against physical dataloss or unauthorised access.

All that has to be handled by the network somehow. As is to be seen the client-server-modell is the most common approach to build and run virtual worlds. But there are some disadvantages (for instance the huge workload on the serverside) which caused researching on virtual worlds based on peer-to-peer plattformes lately.

## 3 Virtual Worlds based on client-server-networks

The client-server-model is the most popular approach to realize Virtual Worlds. For instance, all it takes to join the "Second Live" Metaverse is a client of about 20MB. While peer-to-peer-networks are mostly able to provide a very good loadbalancing, in the client-server-network nearly all necessary processing is performed by the server. The client's main tasks are usually not very network-related, more or less just presenting the server's work, for example by rendering 3D-scenes.

### 3.1 Advantages

Virtual worlds based on client-server-networks are actually very safe compared to P2P-networks. Login-server with static IPs prevent users from various kinds of attacks. A closer look at this problematic will explain this in detail in chapter 4.3 of this report.

Another useful advantage is the easy distribution of software-updates. All clients can receive them nearly at the same time. That helps keeping the network safe, faultless and up-to-date.

Last but not least, the world consistency is easy to keep. Everytime a user makes changes to the world in any way, he transfers the necessary information to the server, where everybody else gets it from later.

### 3.2 Disadvantages

The client-server-model comes along with various disadvantages when dealing with virtual worlds. First of all the world data is stored on the server side. That means in most cases the user can hardly change it, for instance by changing the landscape or build houses. But there are a few exceptions to that rule. Most popular example is "Second Life", where you can buy a piece of land by renting server capacity and do with it whatever you want. Of course that costs real money. It is even possible to rent a whole server, if you buy an entire island.

Another big disadvantage are breakdowns on the server side. If the server fails for any reason, lots of users are no longer able to enter the virtual world. That usually leads to very annoyed customers, because virtual worlds based on client-server-networks mostly already come along with monthly fees for content-updates and server-maintenance.

### 3.3 Sharding

In most virtual worlds thousands of users are acting simultaneously. Possible settings are whole continents or solar systems. To keep the illusion of the virtual world believable, the server has to handle an immense amount of clients, which can be connected from all over the world. Market leader "World of Warcraft" has 2 million active players just in Europe, too much for a single server to deal with. The mechanism to solve this problem is called sharding.

#### 3.3.1 How it works

Sharding can be shortly described as physical partitioning. It is a phenomenon not only in designing virtual worlds. For instance, it is also used within databases like SQL. Unlike usual partitioning where the different segments of the database tables are on the same physical machine, with Sharding tables are segmented across several physically different servers. In terms of virtual worlds that means, that various "copies" of the virtual world are existing in parallel. Obviously this goes along with one very bad disadvantage: users in the world hosted by server A are not able to interact with users from the world hosted by server B. Nevertheless this way of dealing with the high amount of participants and its need for computing power is preferred by almost all popular virtual worlds.

#### 3.3.2 Example: "World of Warcraft"

Market Leader "World of Warcraft" uses sharding very consistently, which is why it is very capable to give a better understanding of what sharding really means in a virtual world. A closer look at the European section tells, that there are actually about 250 so called "realms" for about two million players running in parallel. A complete list can be found under [6]. That makes circa 8000 users per realm. Manufacturer Blizzard Entertainment keeps an eye on how busy these realms are and forbids joining a certain realm, if it is full of people in order to avoid congestion. But unfortunately there is no "local" balancing within a single realm, which means there are still bad lags in highly populated areas like in big cities of the world. However, the realms are physically not connected to each other, which means users on different realms are not able to see or interact with each other. But every realm is running the same virtual world, just inhabited by different people. And the concept of sharding goes even deeper here. Every realm again has physically separated servers for each continent in the world, for so called "instances" (dungeons for very small groups of users separated from the rest of the world), for the users item database, for non-player-characters, for chatting. Sometimes the

user can experience a failure of such a single server, for example if he can not move, but chat.

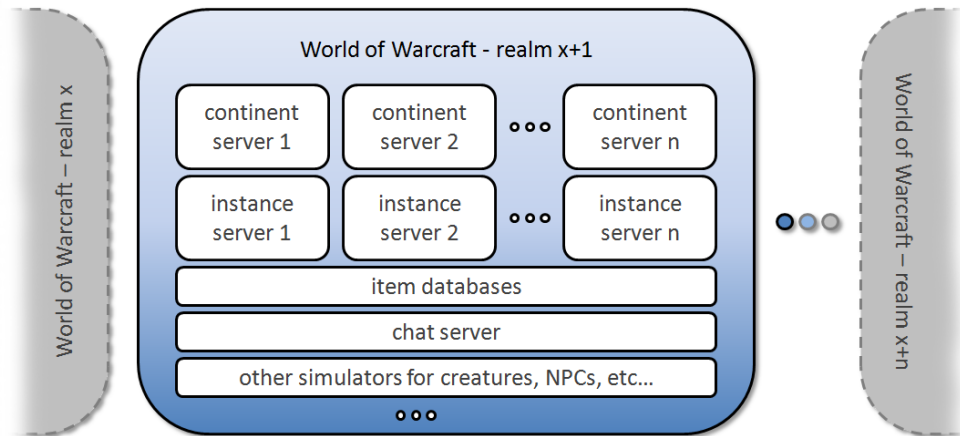


Figure 1: Concept of sharding in "World of Warcraft"

### 3.3.3 How to avoid it

Sometimes sharding is a just insufficient solution, for instance if it makes the scenario implausible. A good example to illustrate that is the MMORPG "Eve Online". This virtual world takes place in a galaxy, consisting of about 5000 solar systems. It is the nature of a galaxy to be very large and to provide space for billions of people. It would be very annoying to split them into groups of maybe 10000 users per server and never let them seeing people of others shards. The game experience would suffer.

In order to avoid that, "Eve Online" makes heavy use of load balancing. All of the computing is done in one single datacenter in London, where a server cluster consisting of 64-bit dualcore-Opteron-based IBM LS20 blade servers is located. Detailed information can be found on [7]. To keep the pace of the processors solid state disks are essential. So every solar system is simulated by a single server. If there are too many people in a single solar system, so that the server can not handle the processing any longer, another server can join helping to provide the needed amount of computing power as long as necessary. So there can be up to 30000 users in a solar system. That is more than enough to handle all users scattered in the galaxy. In contrast to that: a continent server in "World of Warcraft" has its capacitylimit at about 6000 people.

## 4 Virtual Worlds based on peer-to-peer-networks

Peer-to-peer-networks are particularly well suited for communications in loose groups, where the participants come and go in unpredictable intervals. Nevertheless everybody in the network can easily and directly communicate with every other user. Nobody depends on any central authority. At first this looks like a much better approach for virtual worlds, but it not necessarily is.

### 4.1 Advantages

The biggest advantage of P2P-based virtual worlds seems to be quite obvious. There is no server that can fail and exclude maybe thousands of users from the virtual world. The P2P-network is able to administrate itself and a loss of some single nodes is not so serious. Every user hosts his own little piece of land in the virtual world on its own computer. So if his computer fails, just its own piece of land will be unreachable for others. Everything else will work normally. But that means also, if you want to make your own land permanent accessible, your computer must be run 24 hours a day, seven days a week. Otherwise you are forced to use some sort of server, which means it is not pure peer-to-peer any longer.

Another positive side-effect of self-administrative P2P-networks is, that there are usually no monthly fees, because there is just no need for a provider. There are just no servers to maintain and even the creation of new content is done by the community for free.

### 4.2 Disadvantages

Compared to client-server-networks P2P-networks are very vulnerable to different kinds of attacks from the outside, because of not having any kind of central authority. Another problem is how to store informations about groups of users or their privileges. A Problem, that could also be solved by using servers. As can be seen in chapter 5 of this report, a very good approach would be a P2P-network with a very few specialized servers.

### 4.3 Special P2P security considerations

Particularly in P2P-networks it is more difficult to establish trustful relationships than in other kinds of networks. But also in a P2P-based Virtual World it is important to have a trustworthy environment, because like in lots of other applications there can be the need of transmitting passwords or other

sensible data, like bank account data for example, if somebody wants to buy something is a virtual shop. The basic needs, which have to be satisfied are

1. *confidentiality*: If a third person is able to record a message, which was not intended for him/her, this person should at least not be able to read or decrypt the message.
2. *integrity*: Every message has to be received as it was sended. It has to remain entirely unchanged on its way.
3. *authenticity*: The receiver must have the assurance, that the sender really is the person, who he/she claims to be.
4. *authorization*: This is strongly related to (3). It must be guaranteed, that both of the communication partners have the right to do a certain transaction.

A good way to realize all this is the usage of a so called PKI (Public Key Infrastructure). This is a cryptographic system consisting of protocols, services and other standards, which provides the ability to use public key encryption. Every participant gets a unique key-pair, consisting of a private key for encoding purposes (only known to its owner) and a public key for decoding purposes (available for everybody). This concept is called asymmetric ciphering. Digital certificates, handed out by public authorities, can be used to proof the ownership of a persons public key. There is much literature explaining all this in detail (for instance [8], [9], [10], [11]).

#### 4.3.1 models of trust

Trustmodels are struktures that describe, how authority can be passed on within the P2P-network. There are mainly three ways of doing this.

**4.3.1.1 Hierarchic PKI** The first one is a hierarchic PKI with a central certification authority (short "CA") [12]. All participants within a PKI are trusting directly or indirectly (by using chains of certificates) one single CA. This CA is also called root-CA. But this approach might be not very well suited for a P2P-based virtual world, because it comes along with the need for some kind of server for the creation of certificates.

**4.3.1.2 Cross certification** Another approach is the so called "cross certification" [13]. In this model of trust different CAs can certify each other, so this brings a little more freedom compared to hierarchic PKIs with a central CA. Two hierarchic PKIs can be joined together via cross certification. Unfortunately there is a need for a lot of certificate here. A net of  $n$  CAs needs  $n*(n+1)$  certificates. It is hard to keep that consistent, because different CAs can provide different levels of security. Nevertheless this model of trust is better suited for a P2P-based virtual world, because of not having a root CA the participant has the choice to trust who he/she wants.

**4.3.1.3 "Web of Trust"** The third option is called "web of trust". Every participant can create certificates. All it takes is the conviction, that a certain public key belongs to a certain person. In this case the participant signs this public key. Every other user has the free choice now, whether to believe also in this certificate (and therewith in the identity behind a public key) or not. By the time a public key collects more and more certificates. The more certificates are bound to a public key, the more it can be trusted. That is why every new certificate is of disproportionate use for the key. In the economy this is also known as "network effect" [14]. Of course this can be dangerous. A small and malicious group of users can be capable of making a bogus public key popular in no time. A possible counteractive measure could be to establish certain rules for creating certificates or use public CAs for signing public keys, which became very popular. Popularity can be measured by counting the certificates for a certain public key.

So all in all the "web of trust" seems to be the best choice for an independent P2P-based virtual world. Everybody can take care of a minimum level of trust, if he/she follows some easy rules.

## **4.3.2 New approaches to extend PKI**

The bound between a public key and an identity via the usage of certificates is the key-concept in ordinary PKIs. On order to get more options in designing PKIs for P2P-based networks, there are some ways to extend the possibilities. Two of them are SPKI (Simple Public Key Infrastructure) in conjunction with SDSI (Simple Distributed Security Infrastructure).

**4.3.2.1 SPKI** SPKI is still under way [15]. It is based on PKIX and its X.509v2 (certificate revocation lists [16]) and X.509v3 (the actual certificates) standards. The tables 1 and 2 give a short overview on the most important field within X.509v3 and SPKI certificates.

Field	Description
SERIALNUMBER	unique serialnumber, given by the CA that created the certificate
SIGNATURE	Algorithm that was used by the issuer to sign the certificate
ISSUER	identifies the CA that signed the certificate
VALIDITY	period of the certificates validity
SUBJECT NAME	contains the persons name that is supposed to be connected to the public key in the "subject public key info"-field
SUBJECT PUBLIC KEY INFO	contains the public key that is supposed to be connected to the name in the "subject name"-field
EXTENSIONS	for adding additional attributes to the person stated in the "subject name" field. Other informations are possible as well, for instance the CAs security conditions

Table 1: important fields of an X.509v3 certificate

The syntax of an X.509v3 certificate is based on ASN.1 (presentation layer in the OSI-model) [17], a description language for datastructures. ASN.1 is not considerate of computer-intern data representations, which is why the import and evaluation of X.509v3 certificates takes actually very long. A new feature in version 3 are the extensions. There are 16 standard extensions and a lot more private extensions, but both types never became really useful in practice.

The biggest problem with using X.509v3 certificates in virtual worlds would be, that the values in the "subject name" fields have to be unique, which means every username or pseudonym has to be unique. In a P2P-network with no central name lists it is quite complex to make sure that is the case.

So, what can SPKI do better than X.509v3 in a P2P-based virtual world? Compared to a normal PKI authority can be transferred more versatily here. SPKI's "tag"-fields are more flexible than the extensions from X.509v3. They

Field	Description
SUBJECT	hashvalue and hashalgorithm of that persons public key, whose the certificate is issued on, or alternatively just its name or public key
SUBJECT LOC	Address of the subjects public key or a certificate signed by the subject
ISSUER	the hashvalue of the issuers public key and its name
ISSUER LOC	Address of the issuers public key, at least there has to be a certificate which can be used to determine wheter the issuer has the right to create the certificate or not
TAG	for more detailed information related to the given authority
VALID	period of the certificates validity as ASCII-string or data providing to check the validity online
SIGNATURE	the issuers signature, with "DUAL-SIG" can be forced, that the certificate is not valid before the subject has signed to

Table 2: important fields of an SPKI certificate

are able not only to connect public keys with identities, but also to authorize whole usergroups at once (using SDSI [15]) or to authorize for the exclusive usage of certain protocol-types. It is possible to allow a certain number of delegations as well. Furthermore the "subject"-field of SPKI is more flexible than the corresponding "subject name"-field of X.509v3. It can contain not only the subjects name, but also its public key or the hash of this key.

Like in other PKI approaches certificates can become invalid before their period of validity has ended, for instance because of some kind of misuse. Basically this problem can be solved in two different ways. The first one is to check the validity with the help of an online service. This could be realized

with a so called certificate revocation list (CRL) which contains all invalid certificates until their normal period of validity has ended. But this seems to be inappropriate in a P2P-based virtual world, because it requires some kind of central server. It would make the network be dependent on a server. Then it would be more vulnerable, of course. Another approach could it be to limit the period of validity to exact one single transaction. After that they become invalid immediately. This would naturally increase the frequentation of every CA, therewith every participant. It has to be examined, if such an approach is feasible.

**4.3.2.2 SDSI** The Simple Distributed Security Infrastructure adds a few very useful features to SPKI. As already mentioned above values in the "subject"-field must be globally unique. That is not well suited for the usage in a Virtual World. It is very likely that two persons want to use the same name. To make that possible SDSI establishes a new concept for referencing names. It uses namespaces to describe identities. So the names do not have to be globally unique, just "locally". The local namespace of everybodys person is defined by all people this person knows. Their names are called "basic SDSI names". Then it is possible to reference names using other peoples namespaces (so called "compound SDSI Names"). Therefore it is not necessary that the referenced name is globally unique. For example, Alice knows Bob. Bob knows Charly. Therewith Alice has Bob in her namespace and Bob has Charly in his namespace. Now Alice can directly reference to Charly without needing an extra certificate. It does not matter, if Alice has another Charly in her namespace or if there are many other Charlys in other namespaces. To identify this chain of names unambiguously there is the convention in SDSI 1.0 that the origin of this determination is the issuers public key. In other words: Everybody, who is using compound SDSI names in the subject-field when creating a certificate, has to use the own namespace as origin. A compound SDSI name, which uses the issuers own namespace as origin is a so called "fully qualified SDSI name". Many of them can be put together in order to be used as a group. This can simplify a bunch of authorizations.

There is also another important aspect when considering the usage of SDSI for a P2P-based Virtual World: it abstains entirely from the above mentioned CRLs. There are no longer lists of invalid certificates. Instead of that it takes a complete new approach. Every participant, who creates certificates, has to publish an online directory, in which he puts all the information about the certificates belonging to him. Therewith a certificates validity can be checked online. This check is connected to a re-certification,

that means every server, who runs such an online directory, must have to right to recertificate. This can be a problem related to Virtual Worlds. A certificate created by a certain user can only be checked on validity as long as the user is online. Aonther option would be a dedicated server, but that seems not very desirable for a P2P-network.

### 4.3.3 Threats for P2P-networks

Identifying and authorizing each other is just one part in setting up a secure communication. The other one is the transmission of the actual data. Of course this should be done with encryption algorithms, so that informations can not be recorded or manipulated. There are lots of algorithms that can be used, because they have been proven to be very safe, for instance 3DES[18], AES[19]. or Twofish. They can be used in a variety of cyphering protocols in different OSI-model-layers, like SSL(TLS) [20], IPsec and many more. But a Virtual World based on a P2P-network still has to deal with certain threats, it is more vulnerable for, than Virtual Worlds designed as client-server-applications.

**4.3.3.1 Man-in-the-middle attacks** Of course, man-in-the-middle attacks are not a typical P2P-threat, but they are much easier to realize in a P2P-network than in a client server network, because there no fix IP-addresses to connect to, when logging in into the network. So it is not even necessary to spoof a certain IP address. The least to be done is an asymmetric key-exchange (like Diffie-Hellmann[21]) in combination with asymmetric encryption. That can be easily done with SSL/TLS or IPsec. The only danger now is the possibility that the man in the middle took his chance already even before the communication process has started. Then he could have done an asymmetric key-exchange with both of the communication partners seperately. So there is the need for a guarantee that a public key really belongs to the person that claims to be the owner. Therefore (SPKI-)certificates can be used. These certificates must be trustworthy. So they need to be signed by a trustworthy CA. The question is, how this CA can be involved into the P2P-network. If it is designed as "web of trust" there should be no server, which makes it very difficult to integrate a trustworthy CA. So it is still a problem to make an P2P-network without hierarchy completely safe.

**4.3.3.2 Incorrect Routing Updates** Another threat in P2P-networks are incorrect routing updates. They are often prepairing man-in-the-middle attacks. It takes advantage of every node in the network not knowing the complete network topologie, just a few neighbours. Therefore a routingtable

is used, which has to be updated permanently. A malicious node could provide wrong informations to his neighbours. Future requests could be forwarded to the wrong target and valuable informations can be collected, for example passwords or bank account data. Another scenario is to forward a huge amount of data to one single node, that would collapse ("denial of service").

**4.3.3.3 Partitioning** To be as independent as possible the P2P-based virtual world should be as serverless as possible. The question is how to figure out a node, when connecting to the network? A malicious node can lead the new node in a fake network. This kind of attack is called partitioning, because it separates the real network from the faked network. Later the so trapped host sends sensible data to the wrong malicious nodes, which can use them for illegal purposes.

#### **4.3.4 Possible counteractive measures**

As can be seen in the last three paragraphs the most dangerous threat for a virtual world based on a P2P-network is the pretence of a wrong identity or manipulations of data transmissions. It is very hard to eliminate this threat without some sort of central authority. So maybe the best approach is to integrate a few servers in the P2P-network for special purposes, but as less as possible. So which tasks are requiring a server?

First of all and most important for a trustworthy environment could be a registration server, which signs the first public key of every user. This can maybe be connected with a real world service, which ascertains identities, like for instance "PostIdent" in Germany.

Another useful server would be a loginserver. This would hamper man-in-the-middle attacks or partitioning significantly. It could hold a list of users, which are currently logged in and delegate the new host to the topological most appropriate node.

Not a must-have, but indirectly also increasing the security would be an update-server for periodical update checks. The new software versions could be downloaded automatically, which guarantees, that all users have the same security-level.

Last but not least the SDSI-recertification-problem may be solved by an extra recertification server.

Looking at all this different kinds of servers the big question is, if such an amount of servers is not adding more vulnerable points to the network than it is trying to remove. Every new server especially carries the risk of being disturbed by a denial of service attack, for example. Therewith the

advantages of a P2P-network are gone. But it should be also considered that all server elements in the network can be made more reliable by making them highly redundant.

## 4.4 Frameworks for virtual worlds in P2P-networks

The tendency to implement virtual worlds as P2P-based networks is also shown by many currently developed platforms designed like a framework to build virtual worlds as P2P-network. Some of them are introduced briefly in the next few paragraphs.

### 4.4.1 NICTA

The NICTA research project [22] explores the possibilities of P2P-based networks not only related to Virtual Worlds, but also related to wireless handheld networks and scientific purposes.

Its Virtual Worlds technology is one of the most sophisticated approaches. It enables the creation of highly scalable Virtual Worlds. It allows a large number of users to join a virtual space without the need for expensive server-farms. A discovery module takes care of the spatial information of all the objects in the 3-D virtual world and indexes that information on to a peer-to-peer network using distributed areal indexing techniques. The technology also comes along with an interaction service that ensures the different users are able to interact with each other very efficiently. Advanced clustering solutions provide updates arrive in real-time. So this peer-to-peer architecture ensures that the technology can scale to support just ten or ten million users in the same quality. The network engine has a well defined network API. Its primary purpose is to simplify the use of the network engine by application developers with a minimal but flexible set of operations.

### 4.4.2 Vast

VAST (Voronoi-based Adaptive Scalable Transfer [23]) is a network library for P2P-based virtual environments, for example MMOGs [24]. It is based on the research of Voronoi-based Overlay Network (VON) published originally at the 2004 ACM SIGCOMM workshop Netgames [25], and later in IEEE Network [26].

As already mentioned above, the current MMOGs are working with client-server / server-cluster architectures that have scalability limits and are hard and expensive to maintain. VAST provides an open source alternative to create scalable, cheaper, easily deployable virtual environment applications.

The fundamental concept of Vast is a new algorithm to find neighboring nodes and maintain neighbor relationships in a P2P environment. This will be addressed by the VON algorithm [25], that addresses the main problem: finding the right "neighbors" in a P2P environment.

VAST is a dynamically linked library in C++ and Java that provides two simple functions: a node may join the P2P network by specifying a 2D-coordinate and a visibility radius in the virtual environment. In return, VAST provides up-to-date information on the position coordinates and visibility radii of neighboring nodes within view. Further use of the knowledge about neighboring nodes is left to the application developers.

Another nice aspect of VAST is its use of existing open source software whenever possible.

#### 4.4.3 Solipsis

Solipsis [27] is an open source system for a massively multi-participant shared Virtual World designed at France Tlcom Research and Development Labs. It provides the infrastructure for a public virtual territory. Relying on a peer-to-peer architecture, the Virtual World may potentially be inhabited by a theoretically unlimited number of participants.

Solipsis tries to provide a Virtual World which is as independent as possible from the influence of private interests like server ownership. To achieve this, it is based around a peer to peer model with no servers at all. Everything depends on the endusers computers. Additionally, it aims to give users more flexibility in designing interfaces and content in their individual segments of the virtual world. There is no content provided by the creators of Solipsis. Every user starts from scratch.

## 5 Conclusion and personal thoughts

The variety of technologies and mechanics behind Virtual Worlds is as versatile as the worlds they are designed for. More and more people become participants in Virtual World of some kind everyday. A lot of people just play, other ones focus on social networks. In the future Virtual Worlds will serve as business-plattforms, which makes the security more important than ever before. This examination has taken a closer look at the two commonly used network-approaches, client-server-based and peer-to-peer-based networks. At the moment the majority of Virtual Worlds is using the client-server-concept. But the need for networks able to deal with more and more people is strong. This tendency causes research on P2P-networks. These networks are better

scalable. The more people are joining a P2P-based Virtual World, the more processing power is needed and brought by exactly this people. Huge low-cost-networks can grow without requiring maintenance. On the other Hand today's Virtual Worlds based on P2P-networks are still not safe enough to provide important transactions like the transmission of bankaccount-data. But with (S)PKI and SDSI and just a few servers for example, it seems to be possible to create an trustworthy P2P-environment for Virtual Worlds. So even more types of Virtual Worlds can be expected besides gaming and social networks. Virtual shopping malls, major international events or other business conventions and more is imaginable.

## 6 References

- [1] Neal Stephenson, Snow Crash. *Bantam Spectra* , New York, September 1992
- [2] <http://www.wow-europe.com>, (last visited: 30.9.2008)
- [3] <http://secondlife.com/>, (last visited: 30.9.2008)
- [4] Bruce Woodcock. An Analysis of MMOG Subscription Growth <http://www.mmogchart.com>, April 2008
- [5] <http://www.mysql.com/customers/customer.php?id=226>, (last visited: 30.9.2008)
- [6] <http://www.wow-europe.com/realmstatus/index.html?locale=engb>, (last visited: 30.9.2008)
- [7] <http://www.eve-online.com/pressreleases/default.asp?pressReleaseID=25>, (last visited: 30.9.2008)
- [8] Symeon Xenitellis, The Opensource PKI Book - A Guide to Opensource and PKI Implementations, July 2000 (see also <http://ospkibook.sourceforge.net>)
- [9] Alfred J. Menezes, Handbook of Applied Cryptography, 2nd Edition, CRC-Press, Florida, 2008
- [10] Mollin, Richard A. An Introduction to Cryptography, 2nd Edition, Chapman and Hall CRC, Boca Raton, 2007

- [11] Tanenbaum, Andrew S. Computer Networks, Fourth Edition, Pearson Education International, New Jersey, 2003
- [12] S. Kent, Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management (RFC 1422)
- [13] Klaus Schmeih, Cryptography and Public Key Infrastructure in the Internet, *Verlag John Wiley and Sons Ltd*, West Sussex, England, 2003
- [14] <http://www.marketingterms.com/dictionary/network-effect/>, (last visited: 30.9.2008)
- [15] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen. RFC 2693: SPKI Certificate Theory, <http://www.ietf.org/rfc/rfc2693.txt>, September 1999
- [16] C. Ellison. RFC 2692: SPKI Requirements, <http://www.ietf.org/rfc/rfc2692.txt>, September 1999
- [17] J. Larmouth. ASN.1 Complete, first edition, Morgan Kaufmann, Orlando, Florida, 2000
- [18] H. Handschuh, B. Preneel. On the Security of Double and 2-key Triple Modes of Operation, published in L. Knudsen, Ed., Fast Software Encryption, vol. 1636 of Lecture Notes in Computer Science, S. 215-230, Springer-Verlag, 1999
- [19] J. Daemen, V. Rijmen. The Design of Rijndael: AES. The Advanced Encryption Standard (Information Security and Cryptography) first edition, Springer-Verlag, Berlin, 2001
- [20] S. Blake-Wilson, M. Nystrom, J. Mikkelsen, D. Hopwood, T. Wright. RFC 3546: Transport Layer Security (TLS) Extensions, <http://www.ietf.org/rfc/rfc3546.txt>, June 2003
- [21] W. Stallings. Network and Internetwork Security Principles and Practice, Prentice-Hall, New Jersey, 1996
- [22] <http://nicta.com.au/research/projects/peer-to-peer/>, (last visited: 30.9.2008)
- [23] <http://vast.sourceforge.net/>, (last visited: 30.9.2008)

- [24] Shun-Yun Hu, Shao-Chen Chang, and Jehn-Ruey Jiang, "Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games," in Proc. 4th IEEE Intl. Workshop on Networking Issues in Multimedia Entertainment (NIME), Jan. 2008
- [25] Shun-Yun Hu and Guan-Ming Liao, "Scalable Peer-to-Peer Networked Virtual Environment," in Proc. ACM SIGCOMM 2004 workshops on NetGames '04, Aug. 2004
- [26] Shun-Yun Hu, Jui-Fa Chen and Tsu-Han Chen, "VON: A Scalable Peer-to-Peer Network for Virtual Environments," IEEE Network, vol. 20, no. 4, Jul./Aug. 2006
- [27] <http://solipsis.netofpeers.net/wiki2/index.php/Main-Page>, (last visited: 30.9.2008)