# Cheat-Prevention and -Analysis in Online Virtual Worlds

Sabine Cikic
Technische Universität Berlin
Center for Multimedia in Education & Research (MuLF)
Berlin, Germany
cikic@math.tu-berlin.de

Sven Grottke
University of Stuttgart
Center of Information Technologies (RUS)
Stuttgart, Germany
sven.grottke@rus.uni-stuttgart.de

Fritz Lehmann-Grube
Technische Universität Berlin
Center for Multimedia in Education & Research (MuLF)
Berlin, Germany
lehmannf@math.tu-berlin.de

Jan Sablatnig
Technische Universität Berlin
Department of Mathematics
Berlin, Germany
jon@math.tu-berlin.de

## ABSTRACT

Virtual environments and online games are becoming a major market force. At the same time, the virtual property contained in these environments is being traded for real money and thus attains a real value. Although the legal issues involved with this virtual property have not yet been decided, they will have to be soon. To protect virtual property, virtual environment systems will have to conform to certain requirements. We analyze what these requirements are in order to either prevent cheating or at least prove a digital offense has transpired. Along with greater security, this will also reduce the cost of support, which is one of the major cost factors for online games.

## Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*; H.2.7 [**Database Management**]: Database Administration; K.8.0 [**Personal Computing**]: GeneralGames

## General Terms

Virtual Environments, Security

## Keywords

games, cheats, fraud, virtual property, virtual economy


*e-Forensics 2008*, January 21-23, 2008, Adelaide, Australia.


## 1. INTRODUCTION

User numbers in online computer gaming are rapidly increasing, currently turning over roughly $5 billion per year and estimated to increase to $10 billion per year in 2009[9]. The games themselves have transformed from simple adventures into complex online worlds with millions of users and sophisticated virtual economies. These economies have spawned a secondary market in the real world, trading in game items and characters, or even allowing direct conversion between real and virtual money. This real money trade alone has an estimated volume of $2 billion per year and rising rapidly[17]. As the real money value of virtual property increases, related crimes and illicit activities in various forms are also becoming a serious issue. On the one hand, in-game cheating allows some people to illegitimately aquire virtual property, which is later turned into real money. On the other hand, fraud in various forms is becoming a serious issue for both players (as they become victims of such a fraud) and game providers (because of increased support requests). Also, more and more lawsuits are being filed regarding virtual property. Decisions regarding cheating and fraud which used to be in-house to the game providers must become much more reproducible and provable. At the same time, user numbers and therefore cheating and fraud related support requests are expected to increase rapidly, therefore such decisions must become easier and faster to do in order to limit the cost of support. Our analysis of cheat- and fraud-prevention, -detection, -proving pays special attention to these two factors.

Since virtual property is a relatively new phenomenon, there is as yet no common policy of dealing with it and the associated problems. We argue in chapter 2 that sooner or later many companies will accept virtual property to be equivalent to real property and will therefore be forced to protect it much better than it currently is. In chapter 3 we talk about other technical and social requirements on future virtual environments and games. The enhanced protection necessary for legal certainty does not come easy or cheap (which is one of the main reasons why many companies are reluctant to accept virtual property as real). We analyze what would be necessary for this in chapter 4. Finally, there is a short summary in chapter 5.

## 2. THE REAL VALUE OF VIRTUAL PROPERTY

The question may come up how relevant the trade of virtual property really is. Does it pay off to invest real money and programmer time into the protection of non-real objects? Though we would also argue for the ideal value of virtual goods simply because many real persons have interest in them, we shall only argue here in terms of money.

Several efforts have been made recently by designers of virtual worlds to support the trade of virtual goods for real money. One extreme case is *Sony*'s *Station Exchange*[26]. Via this service, users exchange game items for US$ in a protected environment. US$ 1.87 million have been transacted there in the first year[23]. Since users pay for the service – the provider retains a provision – this shows the demand for protection in virtual economies.

Another example is *Second Life*, where the virtual economy is backed by a bank of issues. The exchange rate between its virtual currency *Linden Dollar* appear like "real" exchange rates at the news posts of agencies like *Reuters*, where it is rated as stable currency.

Also disputes about virtual property begin to reach real world judiciary. Some cases on virtual property were already decided in China[5]. In May of 2006, the provider of *Second Life*, *Linden Lab*, rejected a claim by attorney M. Bragg, for compensation of a loss of virtual property he had invested US$ 2000 in, before a legal court of the United States. Interestingly, the accepted amount in dispute – US$8000 – was the estimated market value of the property[6]. Though the heavy public discussion for the case in- and outside of the *Second Life* community saw much more sympathy for the party of *Linden Lab* in the special case, *Linden Lab* eventually settled by fulfilling all demands[21]. Appreciating the 17-month reasoning of some of the most influential experts in that field, the result can be read as: A priori, virtual property IS (natural) property in an economic and political sense.

The volume of the economies in virtual worlds is estimated from around US$500 million[9] to US$ 2 billion[17] per year and thus already exceeds that of the smallest national economies. But this is only the "tip of the iceberg" of its potential, because most massively multiplayer online games (MMOs) not only do not support real money trade (RMT), but explicitly try to hinder it to a certain extent. E. g. an extra feature of the popular MMO *World of Warcraft* (WOW) and others is the so-called "bind on pickup", where "bind" means the invalidation of a virtual item for (RMT). The object has the attribute, that it can never be owned by another subject than the first owner, hence it cannot perform as counterpart of any other change of ownership, particularly that of money. Such a feature would not have been invented without the existence of a drive towards "realization" of virtual property.

Whether in the future we will see the persistence of the somehow paradise-like status quo of non-commercial game worlds or the economic fusion of virtual worlds with the real one – that is not the question. In fact, it is probable that most systems will position themselves across the spectrum between, with a tenacious virility towards the fusion. But in any case the issue will have a growing impact onto the character, the role and the relevance of virtual worlds. In consequence, it is clear, that substantial resources will have to be raised to install some "safety" infrastructure against anarchy in virtual environments. Hence cheat and fraud can be defined as crimes.

Responsibility and power for rules and laws on virtual environments are still shoved and pulled between real society and virtual worlds. In the Bragg vs. *Linden Lab* case stated above, the judge ruled the terms of service (ToS) non-binding[21], and the ToS were since amended to give users more rights (specifically, virtual arbitrations)[21]. Afterwards a settlement was made[21], but no judgment was spoken.

Governments should apply real-world laws and regulations to virtual currencies in online worlds like *Second Life* to prevent potential money laundering, fraud and tax evasion[22].

## 3. THE FUTURE OF ONLINE WORLDS

### 3.1 The Present

Virtual environments have been around for only about fifteen years, *DIS* being one of the first to be standardized in 1993[18]. Even though room- and space-based approaches are often thought to enhance usability[13, 14, 4], virtual environments have not yet found wide recognition and circulation. The reasons for this are unknown but may have to do with the technical issue of very limited interactive possibilities when running a virtual environment over a network with noticeable delay[24]. The implementations available for Internet use today allow the users (or, rather, their avatars) to move around virtual environments, perform simple activation interactions and chat. For the most part, however, these options exist outside of virtual environments, as well. In particular, more complicated interactions like performing a virtual experiment, building a virtual machine, or holding a virtual conference with many users are not possible. Currently, most available applications, e. g. common learning management systems, support a few hundred users at the same time[19], but performance is bad when many users cluster at the same virtual location. One of the problems of virtual environments is that they traditionally rely on a client server model. Therefore, an expensive server has to be provided, which is a problem because most Internet services are not paid for but free.

Computer games, however, have adopted the online model with a lot more zeal. Persistent game worlds where player actions have a permanent effect, complete with a virtual economy of items and services, and a group of regular visitors have existed since the 1980s in text-oriented multiuser dungeons (MUDs). User numbers increased enormously when the games added a graphical front end. *Ultima Online* started in 1997 and is still running with some success today in 2007. *World of Warcraft* started in 2004, has over 9 million subscribers in 2007, and has become the most successful computer game to date[20]. The technical abilities of these games have, however, not improved in the last 10 years. The user's avatar can move around the game-world and perform simple interactions with AI-controlled enemies and other players. In general, players do not have real influence on the virtual surroundings, cannot fell arbitrary trees, for instance, or build a bridge at a random location. There are also no complicated or heavily time-critical interactions as are the norm for real-time strategy games or first person shooters. The reason the interaction bandwidth and reactivity are limited in this way is once again of tech-

nical nature. With the current properties of the Internet, the user experience becomes inacceptable if too many interactions are possible/critical. All current online games use a client-server model, where all users of a particular game connect to a specific server, which handles the actual game. The downside of this is that the server is a bottleneck and cannot handle an arbitrary number of users. When more users are to be accommodated, a new *shard* is added, this is another server with a copy of the game. Users from two different shards cannot see, communicate or interact with each other through the game. The game *World of Warcraft*, for instance, has 800+ shards. Current online games support a few thousand users on a single server simultaneously[27]. The computers used as server engines are usually top of the line to accommodate the most users possible. The computing power necessary to participate in one of these games is negligible in comparison to the complexity of the graphics rendering required.

## 3.2 Future Conditions

Of course the Internet will improve, as it has (vastly) improved in the past. In particular, available bandwidth will improve by many orders of magnitude when optical fiber is brought to the end-user[11]. The *delay*, i.e. the time it takes to deliver an Internet package to the target site, on the other hand, will not improve much further, as it is already fairly close to the theoretical minimum. The speed of light dictates that sending a piece of information from Europe to Australia and back, for example, cannot take less than around 100ms – if the cable were running in a straight line. The real ping-time in 2007 is around 300ms.

Computing power will also increase noticeably, probably following Moore's law (doubling around every two years). Without changing the architecture, simply increasing the server's computing power will at best allow increasing user caps for online games by a similar factor.

## 3.3 Object Complexity

The interactivity of online games should be increased in a similar way as it has increased in offline games. Where ten years ago, offline game interactions were about what they are now in online games, today's offline games often boast completely free interaction and physical modeling. Computer games have always striven to simulate the real world as much as possible (with the addition of a few key differences). To become a true virtual environment resembling the real world, the object density (indicating both amount and complexity of the objects) has to increase by many magnitudes. E. g. a steam-engine should not just be a steam-engine-look-alike-box with an on-off switch but rather should be made up of cylinders, which in turn are made up of nuts and bolts, which in turn are made up of metal, which can be bent or perforated.

Most types of traditional consistency schemes, such as server-client and also loose consistency in distributed architectures, require sending frequent updates of object state to the other hosts. It follows that network bandwidth increases as the amount of objects and the size of their object state increases. Also, with higher object density, there are more chances for disagreements between the hosts or the clients and the server. Even if a common end-result is somehow arrived at, the visual consistency will suffer if many objects are in use. In particular, physical models do not perform realistically under these schemes.

One way of solving the issue is by greatly increasing bandwidth use of the game or simulation. While this will allow increasing object density by a few magnitudes, the increase is limited as the available bandwidth will always be limited and the object density demand will increase much faster than the Internet bandwidth. Also, if a server is involved at any point, either because a traditional server-client architecture is used or even if just to provide logging or security monitoring, the Internet connection bandwidth of that server will increase accordingly, which can become a cost factor.

An alternative is *optimistic consistency* with determinism. In this model, each participating computer simulates the world on its own, including any side-effects and inter-object interactions. In a sense, each host simulates the game like an offline game. As long as the world is left alone, it will play out exactly the same on all hosts. The only non-deterministic information that enters the world and has the power to affect its outcome are user input and a few random decisions. These nondeterministic events, and only these events, need to be communicated to the other hosts. The network bandwidth is then entirely independent of object density and only depends on the number of users. Since the packages are delayed when transferring across the network and may arrive at different times at each server in the network, packages with relevant information may arrive after the relevant action has passed. In order to keep all hosts congruent, each host must be able to backtrack its state to a previous time, apply the late user interaction information, and simulate back to the current time. This also means that the computing power necessary for this system is higher than with the traditional model. The increase is usually a factor of around four[24]. This should not be a problem with current computers, much less with those available in the near future.

Optimistic consistency not only allows bandwidth independence of object density, it also shows significantly better effective consistency[24]. This in fact allows physical modeling, even greater object density, and a better visual consistency. Finally, a complete record of all *nondeterministic* events is sufficient to log and replay the entire playout of the simulation. In particular, hosts may send all recorded nondeterministic events to a logging server, which, in combination with regularly scheduled state-copies, can then be used to document and prove any event in the game, if one has the virtual spatial and temporal coordinates. In other words, an exact replay is possible, which can be a key advantage in security-sensitive environments (see chapters 4.1, 4.2.3).

A virtual environment allowing a high object density and physical modeling would also find applications outside of games, most namely in education.

## 3.4 Scalability

Although online games are already a success, the problem of limited user numbers and low interactivity is a constant issue. The next generation of online games is hoped to support hundreds of thousands or millions of users in a single shard to really allow everyone playing the same game to play it together. It is infeasible to run such a large simulation on a single computer, no matter what calculative power it may have.

To allow a simulation of such dimensions, it then becomes a necessity to split it among several hosts, the simulation is then called *distributed*. Each participating host only replicates part of the entire world, with some overlap between hosts perhaps. This *partial replication* approach allows reducing computational complexity per host from O(N) (where N is the number of users) to O(D) (where D is the Density of users). This would allow scaling to any size.

The architecture of a distributed system can still remain server-client, just that there is no single server but rather a server farm. Note though, that there will be extreme demands on the Internet connection bandwidth of such a server farm. Another possibility are super-peers, where the cooperating servers are distributed across the Internet. The disadvantage is having to operate a multitude of different servers across the world, which is usually not feasible[10].

Another way to achieve scalability is to harness the computing power of the client machines. The game then runs on a peer to peer architecture. This architecture is used both to disseminate messages efficiently and to distribute world state calculation. To date, no working version of a peer to peer online game exists, but studies are progressing rapidly[3, 16, 25]. Note that peer to peer systems do increase the chance for cheating by rather a lot. Note that when a peer to peer simulation is run by a company, it will have to take extra steps to bind the game under its control. This can be achieved by having a single-sign-on point, for instance.

At the same time, moving to a peer to peer architecture does allow installing a virtual environment without the need of a high-end server, which in turn should allow more applications for this paradigm in education, science and every-day Internet usage.

## 3.5 The Catch

Systems employing partial replication are well-studied and they work, but there are some difficulties with these, even without also using a peer to peer approach.

The standard consistency algorithm for distributed simulations is *loose consistency*. In this scheme, each host sends object updates to the other hosts on a regular basis. When object updates from other hosts arrive, it overwrites its own state for the objects in question. The scheme is usually equipped with some additional algorithms such as *dead reckoning* which covers for missing messages and player-ghost estimation which causes additional updates when they are probably needed. While this scheme does work surprisingly well and stable, a particular problem exists where several truths about a single event exist. The reason for this is associated with the concept of serializability (or the lack thereof)[2]. Simply speaking, several conflicting actions can occur at the same time. The conflict resolution then overwrites the states of some of the objects on one host with the view of those objects from another host and vice versa. While this does lead to a common view of the final result, the hosts may have had differing views in the meantime, their history (or log) may differ. Also, the final result may often not be arrived at by any single-host simulation. In other words, the internal laws of the simulation may have been bent. This obviously has implications on disputes. In fact, not only will this effect cause disputes because two people see an event differently, with both of them thinking they are right. It will also be very difficult to settle who is actually right – both of them may, in fact. Also, this bending of the rules almost precludes installing complex rules and hence, physical modeling.

On the other hand, when one tries to combine optimistic consistency and any partial replication, the problem is this: In order for all simulations to play out the same way, one must take into account all influences that can have an effect on a particular object. With partial replication, each host only simulates a subset of objects. If onbject is influenced by an object that the server does not replicate, the simulation becomes erroneous. Even though no published solution for this problem exists yet, we are confident it can be solved and are currently running experiments to investigate it further[24].

# 4. THE PRICE OF SECURITY

As far as virtual property is concerned, the two main virtual crimes are fraud and cheating.

## 4.1 Fraud

Fraud is any kind of deception to gain virtual property from another without paying for it fairly. Con artists have been around since the dawn of time and they exist in virtual economies as well, so the problem is not exactly new. Nevertheless, the problem has been largely ignored up till now, but must be dealt with soon, for the following reasons.

As virtual property becomes more valuable and it becomes ever easier to turn it into real money, real-life criminals start to become interested in virtual economies in the same way they are interested in virtual banking and credit card fraud. Professional criminals will raise the bar on cyber crime and the situation will become a lot more dangerous online.

Also, the way trading works in many online games encourages defrauding your trade-partner. For example, a user wishes to sell a virtual item. He finds someone interested in buying the item outside of the game, e. g. via *eBay*. Inside the game, he trades the item to that person. Then if that person fails to pay in the real world, it will be very difficult for our user to prove in a real-world court that the has real claims because he cannot prove he delivered the item. Accurate logs are not being maintained and there is usually no automatic cash vs. item exchange. Depending on the item or service that is being traded and the game in question, trading may also require sharing identities or passwords, which is obviously very dangerous.

Finally, whenever a user is defrauded of something, he will usually turn to customer support for his game and try to get the property he lost returned. This causes a flood of support requests, which, in turn, cost real money to the company distributing the game[8].

Note that the problem of fraud is not restricted to trading into and out of the virtual environment. There are also plenty of in-game frauds possible, usually when trading. Frauds like switching an item to be traded for a worthless look-alike would currently not be triable in front of a real court, again because the exact circumstances cannot be determined.

There are some rather efficient solutions to the problem of fraud. *Sony* has installed its *Station Exchange Service* which is an out-of-game web site allowing players to safely and securely trade *Everquest II* items to real money[26]. *Sony* effectively provides an escrow service with each user only interacting financially with *Sony*'s automated marketplace

web site. The setup completely eliminates all possibility of fraud during the exchange. Note that Sony's support cost has gone down significantly on the shards where station exchange is in use[23]. Similar setups exist for micro-payments on various other games and environments. Of course people can still cheat or hack the web site or the exchange programs (as happened, e. g. in [6]), but this is beyond the scope of this paper.

It is more difficult to combat fraudulent behavior inside a game. Although the in-game trading interface is meant to protect players by going through several steps when trading, there are many cases where the interface cannot be used, for instance when trading in-game money vs. in-game services like an escort to a difficult-to-reach place or when an agreement exists about which of the players gets which rare item when and if it is found. Once the value of the virtual items exceeds a certain amount, players will usually turn to user support in the case of a dispute. Such disputes cost the company real money, as they have to hire support personnel.

In order to settle the dispute, it is often necessary to know the exact circumstances of the dispute. To this end, keeping a log of all chat between players is helpful, but usually not enough. For a complete understanding, one would need chat-logs, lists of trades and their item lists, even lists of all items picked up by players, all with timestamps. Even with all this data, it would be hard (meaning, time-consuming and therefore expensive) to reconstruct the exact situation which lead to the dispute.

If one wants to keep support cost low in these cases, one has to provide an easy-to-use and all-encompassing log retrieval and conditioning tool. Optimally, this would include a replay of any space and time in the past of the virtual environment. Even poorly trained personnel would then be able to understand what really happened during the disputed event and be able to decide which of the disputing players was in the right.

Additionally, the possibility of an exact replay would be acceptable as proof, which would provide legal coverage. In fact, the feature of swift and reliable dispute arbitration this way could be a selling point for a virtual environment, in the same way that a country with high *Corruption Perceptions Index* draws more investment in the real world[28].

As mentioned in 3.5, there is a particular problem that arises when loose consistency is used on a distributed simulation of some sort. If two players had agreed, for instance, that the one who dealt the deathblow to their enemy would get his item, they may very well *both* see themselves deal that stroke. Afterwards, their foe is dead and both claim his treasure. If the companies customer support is called to resolve the dispute, it will be hard to analyze the data, but if analysis succeeds, no winner can be announced. The only *fair* solution would be to clone the item and give one to each of the champions. This, however, is not in the interest of virtual economy, where the source-flow of items and their uniqueness has to be preserved. Consequently, such consistency algorithms should be avoided.

## 4.2 Cheating

Cheating has been a problem in most online (and offline) computer games since they started. Cheating allows players to do things that he should not be able to, e. g. easily winning over a difficult enemy, running faster than should be possible, finding out where a much thought-after item is hidden rather than searching a large area etc. Cheating hurts the virtual economy as some players will be able to produce a certain resource very easily and quickly, while others will still not be able to do so competitively. If cheating happens with enough different resources, players using legitimate means will not be able to earn an income any more and leave the game.

Many methods have been devised to combat cheating, most of these are closely tied to the game and its implementations.

### 4.2.1 Illegal Knowledge

Quite a few cheats actually revolve around knowing things that the player is not supposed to know (yet). Famous examples are the well-known maphacks and wallhacks, where the player sees what happens in areas he cannot currently see, or has not ever seen. These cheats are some of the hardest to detect, prevent, and prosecute, as even a total replay of the situation, with all possible data available, will often not prove that the player cheated, he may just have guessed well.

Protecting against this sort of cheat depends largely on the game architecture in use, but some means exist for each architecture. For traditional client-server systems, the key is limiting information flow as tightly as possible. This includes not sending internal states of unknown objects to players before they should know about these, e. g. not giving out hints about what item is in a chest before opening it. This also includes restricting the area around the avatar which players are informed about, even purposefully hiding some information within that area.

For peer to peer systems, information restriction becomes much more difficult, but is still possible. The most common cheat here is to let the cheater know what other players are doing before the cheater acts himself, with both actions appearing to be simultaneous. This gives players a distinct advantage in direct competition, which is a part of many games. To prevent several cheats involved with temporal pre-knowledge Baughman and Levine developed the lockstep protocol[1]. This protocol hinges on sending out cryptographic hashes of player's actions for the next action round, then sending out the plaintext action when everyone else has committed their action hash. Thus, all players must decide on an action before they can see any other players cleartext action. The protocol was improved several times to improve performance and address even more cheats[7, 12].

Another weakness of peer to peer systems is the possibility to intercept all information about one's surroundings, thus allowing trap-evasion as well as gathering internal game-information (such as which chest has which content). Since the information flows through the peer to peer network, a cheater could be able to intercept it. The key to preventing this is to assign players to be region controllers for areas they themselves are *not* in at the moment. Thus a cheater can only gain information for a part of the game he has no interest in[16, 15].

### 4.2.2 Illegal Actions

When persistent worlds are involved, user actions matter even when no other players are nearby. This opens up a slew of new cheats, usually geared to either cloning items or generating large amounts of items.

To prevent such cheats, it is paramount that player actions

be double-checked and logged at all times. Periodically or when doubt is cast on the validity of a player's possessions, his past actions can then be audited.

Again, the problem becomes significantly more involved when running on a peer to peer architecture. Normally, other players will provide the region control for the virtual area a player is in. If the player somehow manages to be that controller (perhaps because he has two avatars logged into the game), he can then hack his client to let his main avatar do whatever he wishes. The best way to prevent this is to assign *several* region controllers to each region. Each will then double-check the decisions of the lead controller and report any anomalies to the company's cheat surveillance server. Cheating is then only possible by gaining control of all or at least a majority of region controllers assigned to a region[15].

When a cheat does occur and is reported, it would then be one host's word against another's. Thus, to be able to prove one's innocence, it is necessary to keep a record of all actions and decisions sent and received by each host. Also, actions should be cryptographically signed before sending them out to prevent a cheater from forging other hosts' actions. When an anomaly is detected (which could be caused by network problems, for instance), all relevant hosts must send their logs to the central server for automated review. In fact, in order to be able to have complete reviewability, *all* logs should be sent to the central server whenever a player is in the game. Note that arrival of these logs is not usually time-critical, they can be sent via trickle-upload. Also, these logs may not need to be analyzed immediately, keeping them for later reviews may be enough, or random audits could be added.

Nonetheless, if one wishes to install such a scheme for a very large system with millions of users, it is imperative that the logs be as small as possible and carry as much information as possible. In particular, what actually happened in the virtual world should be reconstructible from the logs and not also have depended on the vagaries of the Internet. This precludes relying on loose consistency, since even with the logs of player commands and of what the region controllers decided, the simulation could have taken several different paths, depending on *when* the commands arrived at the other hosts. This would make automatic detection and auditing difficult or at least render a very large class of cheats impossible to find.

Optimistic consistency, on the other hand, by nature only sends out the player commands and region controllers could send these on to other players in the area and to the logging/surveillance server. The possibility for cheating is much reduced if the commands are signed. Actions like "create a new item so-and-so" are also impossible since the optimistic language only allows user actions like "click at X,Y,Z". Also, if the complete log of all actions is gathered, the final playout is independently deducible by an offline server.

### 4.2.3 Review

Traditionally, when a player is caught cheating by one of the tools above, he will simply be banned, if the company has this policy. Banning here means he looses his place and standing in the virtual world, along with all virtual property he may have had.

While this has worked well and is working, there are also many cases in which the wrong people were suspected, perhaps because there was a problem with the Internet connection or because their computer may have been faulty.

Also, as has been shown by [21], people may sue the company in question. So the banning should be able to stand up in court.

To achieve this, companies should have their customer support review each case in which someone is suspected to cheat before actually banning him. For this to be feasible, the review process has to be:

**Decidable:** Enough data must be gathered that it is easy to decide whether someone is cheating or whether there was a single faulty incidence from some other source.

**Provable:** The review should collect proof in a quality that will uphold in court.

**Cheap:** As usual, the company has to pay for support, so the faster the review process works the better.

The best way of meeting all of these requirements is by providing a replay ability as described above in chapter 4.1.

## 5. CONCLUSION

**Table 1: Comparison of Architectures**

| Feature | Server-Client | Peer to Peer |
|---|---|---|
| Scalability | - | + |
| Company Control | + | ○ |
| Replay | + | ○ |

**Table 2: Comparison of Consistency Models**

| Feature | loose | optimistic |
|---|---|---|
| Scalability | + | - |
| Performance | + | ○ |
| Bandwidth | ○ | + |
| Visual Quality | ○ | + |
| Physical Modeling | - | + |
| Single Truth | - | + |
| Conclusive Replay | - | + |

We have discussed the main aspects and requirements of how virtual environments need to be built to face the new challenges of security and control. We have also listed some of the other requirements virtual environments have to fulfill in order to be competitive in the future. The key features necessary to meet all of these requirements are listed in tables 1 and 2. The tables also show the antithetic paradigms in architecture and consistency model and how well they can support these features.

As table 1 shows, server-client systems work well but cannot be truly scalable. If very large user numbers are to be supported, peer to peer systems will have to be used. These can support the other features if necessary.

Table 2, on the other hand, shows that the traditional loose consistency has serious shortcomings, especially in the area of security. Optimistic consistency has significantly better support for most of the aimed-for features. The authors of this paper are currently researching optimistic consistency's one major weakness, failing to be scalable, and hope to report solutions soon.

## 6. REFERENCES


[1] N. E. Baughman and B. N. Levine. Cheat-proof Playout for Centralized and Distributed Online Games. In *IEEE Infocom*, pages 104–113, 2001.

[2] P. A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.

[3] A. R. Bharambe, S. Rao, and S. Seshan. Mercury: a scalable publish-subscribe system for internet games. In *Proceedings of the first workshop on Network and system support for games*, pages 3–9, 2002.

[4] S. Cikic, S. Jeschke, N. Ludwig, and U. Sinha. Virtual Room Concepts for Cooperative, Scientific Work. In *Conference Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2007)*, Vancouver, Canada, 2007. to appear.

[5] CNN (Reuters). Online Gamer in China Wins Virtual Theft Suit. `http://www.cnn.com/2003/TECH/fun.games/12/19/china.gamer.reut/`, Dec. 2003. Last visited: 2007-10-10.

[6] K. Craig. Wired website. `http://www.wired.com/gaming/virtualworlds/news/2006/05/70909`, May 2006. Last visited: 2007-10-10.

[7] E. Cronin, B. Filstrup, and S. Jamin. Cheat-Proofing Dead Reckoned Multiplayer Games. In *Proceedings of the 2nd International Conference on Application and Development of Computer Games – ADCOG 2003*, Hong Kong SAR, China, Jan. 2003.

[8] S. B. Davis. The Cost of Insecurity - Griefing: from Anonymity to Accountability. `http://www.gamedev.net/reference/business/features/griefing/`, 2005. Last visited: 2007-10-12.

[9] Everquest II website. `http://eq2vault.ign.com/View.php?view=columns.Detail&category_select_id=35&id=434`, Apr. 2005. Last visited: 2007-10-10.

[10] S. Fiedler, M. Wallner, and M. Weber. A communication architecture for massive multiplayer games. In *Proceedings of the first workshop on Network and system support for games*, pages 14–22. ACM Press, 2002.

[11] FOLS website. `http://www.fols.org/technology/`, 2007. Last visited: 2007-10-11.

[12] C. GauthierDickey, D. Zappala, V. Lo, and J. Marr. Low Latency and Cheat-proof Event Ordering for Peer-to-Peer Games. In *NOSSDAV '04: Proceedings of the 14th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 134–139, June 2004.

[13] S. Greenberg and M. Roseman. Using a Room Metaphor to Ease Transitions in Groupware. In M. Ackerman, V. Pipek, and V. Wulf, editors, *Sharing Expertise. Beyond Knowledge Management*, pages 203–256. MIT Press, Cambridge, MA, Jan. 2003.

[14] J. M. Haake, T. Schümmer, A. Haake, M. Bourimi, and B. Landgraf. Supporting Flexible Collaborative Distance Learning in the CURE Platform. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[15] P. Kabus, W. W. Terpstra, M. Cilia, and A. P. Buchmann. Addressing Cheating in Distributed MMOGs. In *NetGames '05: Proceedings of 4th ACM SIGCOMM workshop on Network and system support for games*, pages 1–6, 2005.

[16] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-Peer Support for Massively Multiplayer Games. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 1, 2004.

[17] T. Lehtiniemi. Virtual Economy Research Network website. `http://virtual-economy.org/blog/how_big_is_the_rmt_market_anyw`, Mar. 2007. Last visited: 2007-10-10.

[18] M. R. Macedonia, M. J. Zyda, D. R. Pratt, D. P. Brutzman, and P. T. Barham. Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments. In *Proceedings of the 1995 IEEE Virtual Reality Annual Symposium*, pages 2–10, 1995.

[19] Moodle Forum. `http://moodle.org/mod/forum/discuss.php?d=72766`, May 2007. Last visited: 2007-10-11.

[20] R. Parloff. Legal Pad. `http://legalpad.blogs.fortune.cnn.com/2007/06/page/2/`, June 2007. Last visited: 2007-10-11.

[21] A. Reuters. Linden lab settles bragg lawsuit. `http://secondlife.reuters.com/stories/2007/10/04/linden-lab-settles-bragg-lawsuit/`, also `stories/2007/09/18/linden-revamps-arbitration-in-new-terms-of-service/`, and `stories/2007/05/31/judge-rules-against-one-sided-tos-in-bragg-lawsuit/`, Oct. 2007. Last visited: 2007-10-11.

[22] A. Reuters. Uk panel urges real-life treatment for virtual cash. `http://secondlife.reuters.com/stories/2007/05/14/uk-panel-urges-real-life-treatment-for-virtual-cash/`, May 2007. Last visited: 2007-10-10.

[23] N. Robischon. Station Exchange: Year One. `http://www.fredshouse.net/images/SOE%20Station%20Exchange%20White%20Paper%201.19.pdf`, June 2005. Last visited: 2007-10-10.

[24] J. Sablatnig, S. Grottke, A. Köpke, J. Chen, R. Seiler, and A. Wolisz. Adam – a simulator for distributed virtual environments. Technical Report in preparation, TU-Berlin, 2007.

[25] Solipsis website. `http://solipsis.netofpeers.net/wiki2/index.php/Main_Page`, 2005. Last visited: 2007-10-12.

[26] Everquest II Station Exchange website. `http://eq2.stationexchange.com/`, June 2005. Last visited: 2007-10-12.

[27] D. Terdiman. 'Second Life': Don't worry, we can scale. `http://news.zdnet.com/2100-1040_22-6080186.html`, June 2006. Last visited: 2007-10-11.

[28] Transparency International website. `http://www.transparency.org/news_room/in_focus/2007/cpi2007#pr`, Sept. 2007. Last visited: 2007-10-12.